AGOSTO 1985 Nº 5 A 1,50.- REP. ARGENTINA



K64

COMPUTACION PARA TODOS

19 Programas Inéditos

Desarrollos:

Convertimos la TS 2068

Interface de Grabador Para Commodore C64

Segundo Concurso: Importantes Premios

AVANZADA TECNOLOGIA

Los progresos que se están logrando en materia de semiconductores auguran computadoras más veloces, más reducidas y de menor consumo.

En pag. **4**

PROGRAMAS INEDITOS



INTERFACE PARA COMMODORE

Circuito que permite conectar un grabador común a la Commodore 64

En pag. **28**

CARTA DEL DIRECTOR

Aunque parezca obvio decirlo en esta revista, queremos remarcar que las microcomputadoras "hogareñas" sirven no sólo para jugar sino que permiten una gran cantidad de aplicaciones prácticas. Un experto en estas cuestiones recordaba que un equipo de pocos K de memoria era utilizado, no hace mucho tiempo, por el Banco Central de la República Argentina para ordenar las finanzas del país. O sea que a una máquina del tipo de las que nos ocupamos en K 64 se le puede sacar "jugo" y aprovecharla en campos tan diversos como la educación, la contabilidad, la abogacía o la medicina. Ya hemos dado algunos ejemplos, y en esta edición continuamos ofreciendo programas utilitarios. Uno de ellos fue preparado por un juez para evaluar a alumnos de la carrera de Derecho. Otro ayuda a calcular los "sueldos temporarios".

Y en nuestro afán de darle soluciones a los usuarios de computadores, publicamos en exclusividad dos desarrollos: uno soluciona el problema de software de las TS 2068, mientras que el otro sirve para conectar un grabador común a la C 64.

Y muchas otras notas más que nos colocan en el camino que nos hemos propuesto para presentar el mejor material inédito en una revista de nivel internacional.

CRISTIAN RUSSO

CONVERTIMOS LA TS 2068

La mayoría del software ideado para Spectrum no se puede ejecutar en la TS 2068. Pero es posible hacer que esta máquina "emule" un Spectrum colocando al ROM de este último en reemplazo del original.

En pag **20**

COMO SACARLE JUGO A LA CZ Y TK

En pag. **16**

OTRAS NOTAS

K64
COMPUTACION PARA TODOS

AÑO 1 Nº 5 AGOSTO DE 1985

Director General
Ernesto del Castillo
Director Editorial
Cristian Russo
Director Periodístico
Fernando Flores
Director Financiero
Javier Campos Malbrán
Secretaria
Moni Ocampo

Departamento de Publicidad:
Jefe: Dolores Unan
Promotora: Mónica Garibaldi
Departamento de Avisos
Oscar Devoto
Diagramación y Armado
Fernando Amengual y
Carlos Boccardo
Fotografía
Juan José Pérez
Esteban Figueredo

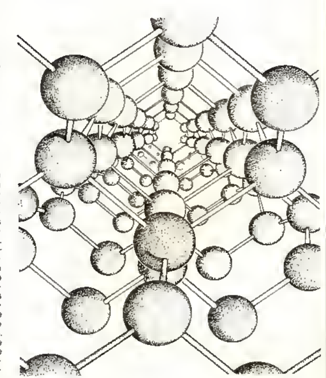
K-64 es una Revista mensual editada por PROEDI Editorial S.A. (e. f.), Carrito 1320, 1º Piso, Buenos Aires, Tel. 42 8881/8 Registro Nacional de la Propiedad Intelectual 313 837 M. registrado Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual.

Precio de este ejemplar: un austral con cincuenta centavos. Precio de la suscripción semestral: 8 australes.
Distribuidor en Capital: Infisito, Venezuela 1417 Capital Federal Tel. 37 8880
Impresión: Calcotam Fotocromotapa Columbia
Fotocomposición: Van Waveren
Los ejemplares atrasados se venderán al precio del último número en circulación.

CORREO ARGENTINO
FRANQUEO PAGO CONCESION Nº 2338
TARIFA REDUCIDA CONCESION Nº 835

AVANZA LA TECNOLOGIA DE LOS SEMICONDUCTORES

Progresos en la tecnología de los semiconductores se están logrando en los laboratorios de IBM, que auguran computadoras más veloces, más reducidas y de menor consumo.



Cristal de silicio semiconductor. Su perfecta irregularidad está interrumpida por un átomo de "impureza".

No es ningún secreto que la tecnología de la computación ha avanzado a un paso enérgico desde la aparición de los primeros modelos. Hoy se puede tener sobre el escritorio mucha más potencia informática que con aquellos "monstruos" de hace unos años. Los modelos de los próximos tiempos serán aún más pequeños, potentes y baratos que los de este año. Y de las máquinas de dentro de diez, deberemos estar preparados para ver milagros. Pero para los investigadores del SST, el departamento de ciencia y tecnología del semiconductor, el progreso no es cosa de milagros instantáneos. En realidad es producto de una larga trayectoria y gran cantidad de conocimientos científicos acumulados. La tecnología computacional del futuro es la mayor preocupación de los hombres y mujeres que trabajan en el SST y mucho de lo que se vea en la próxima década, será consecuencia del trabajo que hoy están realizando.

"Nosotros somos los responsables de poseer los programas más avanzados de investigación en la ciencia del semiconductor", dice John Armstrong, director del SST, y vicepresidente del Departamento de Investigación en Lógica y Memoria. "Es una gran responsabilidad. IBM tiene miles de personas trabajando en la tecnología del semiconductor y en el encapsulado, por eso debemos conocer a fondo su trabajo". Existe un programa que encara el SST y otras secciones para investigar en conjunto qué es lo que se necesita. El Advanced Packaging Technology Laboratory (APTL) y el Advanced Silicon Technology Laboratory (ASTL) son dos buenos ejemplos de ellos. "El trabajar en un programa conjunto hace que se puedan explotar a fondo las fuentes en tecnología avanzada de IBM", afirmó Armstrong. "Además ayuda a la transferencia de tecnología".



Armstrong: 'Trabajo visionario'.



Tareas de inspeccion en la fabricación de chips del tipo VLSI.

NUEVOS DISPOSITIVOS

La investigación en Yorktown está dividida principalmente en cuatro áreas: tecnología avanzada en chips, tecnología del encapsulado; materiales y procesos, y ciencia del semiconductor.

Tecnología avanzada en chips es lo que su nombre indica: la invención y desarrollo de nuevos dispositivos semiconductores y procesos que eventualmente puedan ser usados en computadoras IBM. Los chips resultantes deberán ser juntados en un módulo y los módulos en una plaqueta, que se hallara interconectada, a su vez, con otras y con el mundo exterior.

La tarea del encapsulado por su parte es la que permita hallar los medios de "empacar" los circuitos lo mas densamente posible, de modo de hacer a una unidad simple y rápida a la vez.

Los materiales elegidos para desarrollar nuevas tecnologías y nuevos encapsulados, tienen un enorme impacto en el funcionamiento del producto final, como así también al proceso usado en la producción. "Si se usa el material adecuado y el proceso correcto, la Naturaleza estará de nuestro lado", dice Joseph Logue, asistente de director de APTL de IBM. "Si elegimos equivocadamente, tendremos problemas ..."

MAXIMA POTENCIA, MINIMO ESPACIO

Recientemente, un grupo dedicado al estudio avanzado en tecnología del semiconductor dio un gran salto en esa dirección, en un proyecto que forma parte del programa de gobierno "VHSIC" (very high speed integrated circuits).

Ellos produjeron circuitos en los cuales la menor dimensión es 0.5 micrón. Estos circuitos están hechos a base de tecnología del tipo n-MOS FET. Aunque ya estaba desarrollada esta tecnología anteriormente nadie aun había logrado estos niveles de integración y a tan bajo consumo de corriente.

Otra tecnología que aparezca en competencia es la del tipo CMOS, que resulta en dispositivos más complejos pero de menor consumo. Este mismo grupo también está aplicando su experiencia en n-MOS en el área de los CMOS, para desarrollar dispositivos y memorias de alta velocidad para computadoras.

UN CANDIDATO POTENCIAL: ARSENIURO DE GALIO

A pesar que ahora domina la tecnología basada sólo en el silicio, parece que se ha llegado al techo en lo que se refiere a velocidad y performance con este material. Los investigadores están ahora entusiasmados con otras aleaciones de semiconductores. Entre ellas el Arseniuro de Galio, es objeto de intensos estudios.

El arseniuro de galio ofrece intrínsecamente mayor velocidad que el silicio. Simplemente, los electrones se mueven más rapidamente en él. Se han logrado al día de hoy con tecnología MESFET (metal semiconductor FET), tiempos de conmulación menores a 20 picosegundos! (un picosegundo as 1/1.000.000.000.000 de segundo...) Estos dispositivos además consumen menos corriente que sus equivalentes de silicio. De ahí también su atractivo.

El silicio ha venido siendo estudia-

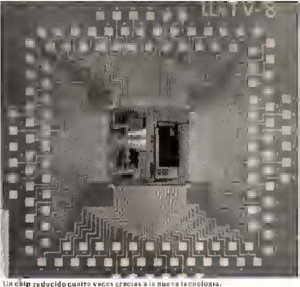
do hace 20 años; el arseniuro de galio en cambio recién está en su infancia

"Básicamente, debemos reinventar la rueda", dice Dean Eastman, director del APTL y jefe del programa sobre Arseniuro de Galio

Seguramente estos dispositivos estarán funcionando pronto en las futuras computadoras, sumergidos en nitrógeno líquido, que es donde se logran las mayores velocidades de trabajo.

Pero el trabajo no es fácil, aún existen muchas barreras tecnológicas que saltar. Todos sabemos mucho sobre el silicio, pero poco sabemos respecto a esta nueva posibilidad.

También se hallan aplicaciones en áreas tales como microondas y la industria de la opto-electrónica. Mientras tanto, los científicos continúan enfatizando ambas tecnologías, exprimiendo aún más las posibilidades del silicio, y avizorando las del arseniuro de galio



Un chip reducido cuatro veces gracias a la nueva tecnología.

GLOSARIO

LETRA "E"

EDIT:
Corrección de programas o parte de ellos.

EDITOR PROGRAM:
Programa que permite la manipulación de texto para su corrección mientras aún se encuentra en la memoria de la máquina.

EMULATOR:
Dispositivo o programa que hace que un ordenador simule el comportamiento de otro.

EPROM: ERASABLE PROGRAMMABLE READ ONLY MEMORY:
Una memoria que puede ser grabada y usada como una ROM y que puede volverse a programar, borrándola previamente con luz ultravioleta.

ERGONOMIC:
Diseño de equipos o accesorios de uso intensivo, de modo que resulten de uso cómodo y descansado, reduciendo así el esfuerzo físico, visual e intelectual exigido al operador.

EVEN PARITY:
Condición que ocurre cuando la suma de bits puestos a uno en un byte, es par (el cero se considera par).

EXCLUSIVE OR:
Operación lógica de Bool. Resulta un uno lógico cuando en cualquiera de los registros de entrada haya un uno (en los dos a la vez no). En cualquier otra situación el resultado es cero.

EXECUTIVE PROGRAM:
Programa o parte de programa que se encarga de controlar la operación general de un programa o sistema complejo

FILE GAP:
Pequeños trozos de cinta sin grabar usados para separar archivos o blocks de datos en un medio magnético.

FILES:
Se refiere esencialmente a blocks de información formando un "record" o grupo de datos. Archivo.

FIRMWARE:
Un programa suministrado dentro de una ROM. Referido a software no modificable, grabado en forma permanente

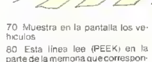
FLAGS:
Celdas de almacenamiento de un solo BIT, que sirven para indicar si existen o no ciertas condiciones lógicas luego de realizadas operaciones en la CPU.



FROGGER

Comp. CZ1000/1500 TK83/85
Conf. 2 K
Clas. Entretenimiento





INSTRUCCIONES:

Esta es una de las tantas versiones del conocido "programa de la rana".

Hay que guiar la "RANA" (F) para que cruce una calle sin ser atropellada. Los comandos son las teclas 5 y 8 para ir a la izquierda y derecha, y la tecla 7 hace avanzar la rana.

VARIABLES:

A$,B$ representan los vehiculos de la calle.
A,B mantienen las coordenadas de posición de la rana.
C$, memoriza el número de la tecla pulsada.
S mantiene el score.

PROGRAMA:

Lineas. 10 a 60: Generan la pantalla en posición de comienzo. INKEY$ es el comando que lee el teclado e indica qué tecla es pulsada.

70 Muestra en la pantalla los vehiculos.

80 Esta linea lee (PEEK) en la parte de la memoria que corresponde a la pantalla la próxima posición de PRINT AT. Si el valor es mayor en ese lugar de la memoria que 127, significará que el auto se superpone a la rana y al programa parará.

90 Borra la ultima posición de la rana.

100 Bifurca a la rutina de avance de la rana si se ha pulsado la tecla "7".

110 Genera las nuevas coordenadas para desplazar la rana hacia los costados, en caso de haber pulsado las teclas "5" ó "8".

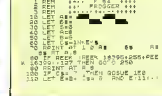
120 Imprime la rana en la nueva posición.

130 a 150 Generan la nueva pantalla y regresa a la linea 60 para recomenzar la rutina.

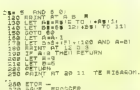
160 a 220 Incrementa "S" si la rana se movió hacia adelante e imprime el score.

PANTALLA





```
  1 REM    +*..   K 64   ++++
  2 REM    ++'  FROGGER  +++
  3 REM    +*....+*..-.+++++
 10 LET A$=
 20 LET B$=
 30 LET S=0
 40 LET A=0
 50 LET B=6
 60 LET C$=INKEY$
 70 PRINT AT 1,0;A$;      B$      A$
   B$ AT A,B
 80 IF PEEK (PEEK 16396+256*PEE
K 16399)>127 THEN GOTO 250
 90 PRINT AT A,B
100 IF C$="7" THEN GOSUB 160
110 LET B=B+(C$="8" AND B<11)-(
```



```
C$="5" AND B>0)
120 PRINT AT A,B;"F"
130 LET A$=A$(2 TO )+A$(1)
140 LET B$=B$(12)+B$( TO 11)
150 GOTO 60
160 LET A=A-1
170 LET S=S+(1)+(100 AND A=0)
180 PRINT AT 12,0;S
190 IF A>0 THEN RETURN
200 LET A=9
210 LET B=6
220 RETURN
250 PRINT AT 20,11;"TE PISARON."
260 STOP
270 SAVE "FROGGER"
280 RUN
```

EXAMEN

COMP.: TS 2068
CONF.: 48 K
CLAS.: EDU



PANTALLA

FACULTAD DE CIENCIAS JURIDICAS

5 marzo 1985

Dr.Luis M.Geibrois

Alumnos del curso de Derecho Procesal II de la Facultad de Ciencias Jurídicas de la Universidad del Salvador, fueron evaluados mediante el uso de una computadora.

La tarea estuvo a cargo del Dr. Luis M. Geibrois — FOTO —, profesor titular de la materia, quien utilizó una microcomputadora hogareña de su propiedad (TS 2068) y un programa en lenguaje BASIC que él mismo diseñara tras varios meses de labor (lo publicamos por separado).

El programa estructura las preguntas que conforman el examen (en esta ocasión fueron 25) de forma tal que aparecen en pantalla numeradas, y conjuntamente con la alternativa de dos, tres o más respuestas, de las cuales tan sólo una es la correcta.

También permanece en pantalla a lo largo de la examinación, un señalador de la cantidad de respuestas correctas e incorrectas. Finalizado el interrogatorio, aparece un "certificado" con el nombre del alumno, y en el que consta el resultado final. Luego, ingresada la calificación, se extiende la constancia respectiva mediante el uso de una impresora térmica (Alfacom 32).

Terminado el turno de exámenes, luego de rendir el último alumno, aparece en pantalla el listado de examinados y las notes respectivas, lo cual equivale, una vez copiado por la impresora, el "acta volante" que confecciona habitualmente todo tribunal examinador.

Y si bien el sistema de preguntes con "respuestas por alternativa" se ha venido utilizando desde tiempo atrás en exámenes escritos, su implementación a través de una computadora le otorga características propias, que lo diferencian por completo de los dos sistemas tradicionales (el oral y el escrito).

A la imparcialidad del examen escrito, se le agrega en este caso la garantía de la "infalibilidad", desde que las preguntas han debido ser necesariamente preparadas con anticipación más que suficiente como para verificar su acierto académico.

No se dá tampoco en la máquina el cansancio del examen oral, que de prolongarse, se torna inevitablemente altamente discrecional.

Por otra parte existe una exigencia intelectual muy severa para el alumno, muy difícil —por no decir imposible— de obtener mediante la evaluación escrita, como es la necesidad de responder a una orden secuencial inalterable. Es decir las preguntas se suceden una tras otra y no pueden "saltearse" como es costumbre actual en los escritos.

Pero, como contrapartide, juega a favor del alumno la posibilidad de regular su tiempo de respuesta, por cuanto el manejo del teclado le pertenece por completo reservándose claro está el profesor la tecla 0 para dar por terminado en cualquier momento el examen con el clásico "suficiente" si lo considera apropiado.

No existen las distracciones propias del examen oral, ni las nefastas influencias visuales.

Pero además de otros aspectos docentes, una particularidad en el examen por computadora garantiza su vigencia en lo futuro: es la obtención inmediata del resultado (situación ésta que no se puede dar en el examen escrito, y muy difícilmente se logrará en un examen oral).

Esta rapidez es la más notoria de sus ventajas, la cual a su vez produce dos fenómenos pedagógicos de indudable repercusión.

Uno se da cuando el marcador de respuestas incorrectas ha ido creciendo a lo largo del examen, de forma tal que es el mismo estudiante quien va asimilando paulatinamente el aplazo, sin tener que sufrir la pequeña humillación, de la más que común inflexión de reproche que se acostumbra formular en el examen oral. O la tediosa espera, a veces de días, cuando no semanas, del resultado de la corrección del examen escrito, que cuando llega desvincula el esfuerzo del resultado, y prácticamente carece de interés.

El otro fenómeno ocurre cuando el marcador de las respuestas correctas le es favorable al alumno. Entonces se advierte que "constituye un formidable acicate para seguir profundizar, ánimo vital éste, que tan pocos profesores insuflan, tan retaceado por otros, como inexistente en los demás", según Gaibrois. "Y la gratificación entonces, nace plena y espontánea en el alumno, que por otra parte la merece, cara a cara con su profesor, porque el estudio, como el trabajo, es el fruto de un esfuerzo ennoblecedor", agregó.

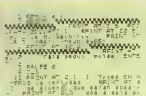
La máquina programada con preguntas, excluye la improvisación en la formulación de las mismas, característica que presentan muchos interrogatorios orales, y si existe repetición ésta no es conocida por el resto de los examinados (que de existir, obliga a una tediosa búsqueda de novedades, no siempre bien encontradas en el momento).

"Por cierto que existen grandes limitaciones en el sistema, fundamentalmente referidas a la examinación conceptual o teórica del estudiante. Por ello se presta muy bien el examen parcial de la materia como en este caso, constituyendo un auxiliar inestimable de la docencia", reconoción el profesor.

"La máquina atrae al estudiante —advirtió—. Y en este caso se ha advertido en la juventud, una especie de renacer del coraje criollo, un deseo de vencer el miedo a lo desconocido y porque no decirlo, un beneficioso sisteme de competir, de 'ganarle a la máquina'."

"Si el deseo de ganar que se genera, se opera a través del conocimiento, no puede haber máquina superior. Se habrá programado para mejorar", concluyó.





```
PALFE 0
CLS
11 PRINT AT 0 1 ; Trees EN NU
MEROS la jentidad   PRINT AT 1
2 de alueno, que daran acceso
- PRINT AT 2,3  ... a esta ev
aluacion
12 INPUT X  PRINT AT 4 10,X
13 PRANT AT 7 4 ; Tipee la f
echa del dia de    hoy (Ro: 1
. 7 Junio 1985).   PRINT AT I.
2
16 INPUT Fa  PRINT AT IO 4 Fa
PRINT AT 1E E
IE PRINT AT . 0 ;3..Ar0'ees
ravue de que idearitada lee pr
eguntas ar cle qria'ando Ud
:teda da' aar ta ai-nado al a/ae
a' P. senao  0  .
IE PRINT AT 4E 0  erifrque.Si
ne +a asurudledo eltrfae' Pulse
er  ENTER ; varias aacrebir,
Fagui   eulie ENTER
17 INPUT ra
18 IF ea1 1r  THEN GO TO I1
20 DIM 4a , 20,
22 DIM C 1
30 LET A10
32 LET A=A,I
4E CLS
49 POKE 23609 4B2
50 REM aU
55 PRINT
60 PRINT AT . 4 4  FACULTAD DE -
IENCIAS JURIDICAS
61 PRANT AT 4 0   PRINT AT
2 3    PRINT AT 1 0
62 PRINT AT 4 34    PRINT AT
... PRINT AT
66 PRINT AT 7 0    PRINT AT
E 0    PRANT AT - 0    PRANT
AT E 0    PRINT AT 9 0    P
RINT AT I0 0    PRINT AT II 0
    PRANT AT 42 0
70 PRANT AT 7 5 E acan aarine
. da   PRANT AT 8,5  Derecho P
cesal   PRANT AT 9 E  Crvrl y
...
... 5 M. Gaibtis
... PRINT AT 20 0
75 PRINT AT 24 0  FLASH I
PRINT AT 2I 4  (A .44  ENTER ).
80 PAUSE 0  FLASH 0
85 CLS
I00 REM Mantar.e Noebre y Apell
ido.
... PRINT
1.0 ... 2  Por fave
PRINT AT - E  ... an al la
0 24  4   PRINT AT 8 3  eaquens
su NOMBRE  ...   PRINT AT
I0 5  Luaeg  ...   ENTER
... PRINT AT  0
1a0 PRINT AT  - 0
... 14 -
180 CLS
180 ... 0a 14 ... 4 ... B.
CHRA 2a CHRa I.
192 LET ... INSTRUCCIONES
19e FOR A=4 TO LEN Fa
IE6 PRINT FN (a 14
I98 BEEP ,I0 20
I9A NEXT I
200 REM  Pantalla Instrucciones
207 PRINT AT 4 5 "A continua..
on aparaceran en    dente la las
 Prasuntas sobre    urs que va'a
a aata exeeer.
ran las reapuestas.
209 PRINT AT 0 I. "Taabran fis.
taaania e  nuearo   Tiaaa corre.
 qua arige coeo     0a ra oarion
oraue no llane     iesaueele  P
ee rorraair eua    poeibrlidad
 20I PRANT AT 14 0   e 10144I
aa aate ordenador   EL tacleoo
 da una eaqurne aa   aa el erado
                     aacriair.
                    "Comarandido
 ecelancal.
299 PRINT AT 20 0
300 PRINT AT 1. 0   Rere comanz
e' pulaa  ENTER r.
31B PAUSE E
320 FLASH 0  CLS
420 LET RA-0
440 LET RD-0
480 LET eN=0
597 PRINT AT . 0
994 PRINT AT 20 0
995 PRINT AT 21 0 " Pulaa nue-
o da opcron y ENTER!
I0SS GO SUB 5300
4100 GO SUB 8000
I502 PRANT AT 1 ;   Art.320 CA:
:500 PRANT AT 5 E    aerrllo 0a
 aeeenae date      :reriee1 al b
onlo ra:laeedr
IE07 PRINT AT 14 I   Sr E1.   PRA
NT AT 1E I  2I NO.
4510 INPUT va
4515 IF va= 0  THEN GO TO 5450
4520 IF v$= I  THEN GO SUB E050
4525 IF va= 2  THEN GO SUB 8200
4E30 IF va > 1  AND va 1 7  THEN
 GO SUB 7900
4540 GO SUB E300
I0E0 GO SUB 8000
ISE5 PRINT AT 7 2, Haale leari
puade el actor   aoarlirer 4 r
aeande ?
4557 PRINT AT Ie I  I) heaaa e'
er de abrirte ar      jurado ?.
piudaa,   PRINT AT 1- I  2  ha:
a antae ee qua  5 eeean.    ee.
aa notificaea
:560 INPUT va
1565 IF va= 0  THEN GO TO E652
1570 IF va= 2  THEN GO SUB E0E0
I5-  IF va= I  THEN GO SUB 5200
I580 IF va    AND va   1  THEN
 GO SUB 7900
IE90 GO SUB 3300
I600 GO SUB 8000
:6I0 PRANT AT 0 I  Er ertio- pueaa
a aealier la cuan-  1ie =e  > 'a:
 abado en la ea-   aenee  er ant
ae ee ra aantan-   1ie vanian nu
evoe periodaS de    e obrrseciro
 7 "
I6I3 PRANT AT I4 7  1  E1.   PRI
NT AT 4B 5, 4  Na,
IEIS INPUT va
IS20 IF v$= 1  THEN GO TO 5450
IE25 IF v$= 4  THEN GO SUB 8050
IE30 IF v$= 2  THEN GO SUB 8200
IE30 IF v$  I  AND ve<r 7  THEN
 GO SUB 7900
I640 GO SUB E300
IEE0 GO SUB E000
I655 PRANT AT 5 I  Por cuanaoa :
:aa aa conflara     e' treelaeo c
a la deeenda en     e  juicio ore
1'e.10
:6E- PRINT AT 44 E  I) par 4B di
ar.   PRANT AT 46 5  2r por 4E 7
raa.   PRANT AT I8 5  3r por 20
dias.
:980 INPUT ra
4E5E IF La- 2  THEN GO TO 5450
4570 IF va= 2  THEN GO SUB E057
4475 IF va= 4  OR va= 3  THEN GO
 SUB 8200
I850 IF ve  :  AND v$  2  AND
va  3  THEN GO SUB 7990
I6B5 GO SUB E300
I-E0 GO SUB 8000
4305 FRANT AT 5 2, 'Art.345 CP
4306 FRANT AT E 2   Si el actor n
e 1iere aoerralir   pero trana in
evebla an  e Fe-   eualrae  for'
CPPEide  e e 'ep-  sron aa eirar
aa.
4310 PRANT AT Ie 5  11 7r.   PR:
NT AT I6 5  2  No.
4315 INPUT vE
4320 IF va= 0  THEN GO TO 5450
4325 IF va= 2  THEN GO SUB 505C
4330 IF va= 4  THEN GO SUB 8200
4335 IF v$  I  AND ve   2  THEN
 GO SUB 7900
--24 GO TO 8E00
7900 PRINT AT 42 0  FLASH 1  TI
REO NAL PAEETE MAS ATENCION!.
7EI5 PAUSE IOE  FLASH 0
7930 RETURN
8000 REM = Sutrutine contador ee
 preguntee.
E0I0 LET PN=PN+I
EEI0 PRANT AT 1 IE  Pregunta
eN
8020 RETURN
8050 REM - Sub ...ie contador de
preguntas correctas.
8055 LET RA=RA+I
8060 PRINT AT 9 0  FLASH 1    C
ORRECTO
8065 PAUSE I70
8070 PRINT AT 0 I  FLASH 0
8075 PRANT AT 0 0; Correctaa
eA
8085 PAUSE 60
8090 RETURN
8200 REM - Subrutina contador de
 respuestas incorrectas.
8205 LET RD=RD+I
8210 PRANT AT 0 I7  FLASH I;   I
NCORRECTO
8215 PAUSE 470
8220 PRANT AT 0 17  FLASH 0
8225 PRINT AT 0 :2  Incorrectas
 RD
8235 RETURN
8300 REM = Subrutina borrado de
entre ... linea 4 a linea 2I.
8304 PRANT AT 4 0
E385 PRINT AT 5 0.
8340 PRINT AT I0 0
8315 PRINT AT IS 0
8.30 RETURN
8459 CLS
8458 PRINT AT 42 40  EUFICIENTE
8458 PRINT AT 49 0
8457 PRINT AT 20 0    Para asegur
.  Pulse  ENTER  .
8460 PAUSE 0
8500 REM - Pantalla   ne
8505 CLS
8510 PRINT
8515 PRANT AT I .  FACULTAD DE C
IENCIAS JURIDICAS
... PRINT AT .
8E2E PRINT AT 1 .  E  4 umn.
4 Ar
8528 PRANT AT - I  ha rendrao er
.a lache examan   Parcral de ..
 asignatura Dera-  cho Procesal
Civil y Com.. ne-
8E30 PRINT AT 40 4  bienao raaaa
10120  .AA   veres en  foraa :
orraaaa  y   RC    vacaa an  /:
rea inrorrerle.
8032 PRANT AI I7 0 :a
8538 PRINT AT I7 I3
8540 PRANT AT IS IJ  Ir..ura M b
eibrole'
8E4E PRINT AT 27 0  1Er.Profeso'
  ingrese le re... ricacror dar
 alueno y ENTER .
8550 INPUT C(A
8551 CLS
8552 PRINT AT 5 2  Er a.rano
Nar(A)
          ha abtenido la calificac
ion
          aa   C(A)
8553 PRANT AT 2: 4  rAara asegur'
  purse  ENTER !.
8558 PAUSE 0
8560 IF A x THEN GO TO 35  CLS
8565 CLS
8570 FOR U=I TO x
8580 PRINT Na ;   C(Ur
8590 NEXT V
```

PANTALLA



CARRERA DE RATAS

COMP: CZ1000/1500 TK83/85
CONF. 16 K
CLAS: ENT

Este juego sólo ocupa 4 K de memoria RAM. Cada jugador posee una laucha y 200 pesos para apostar (usted sólo puede apostar a su rata).

Si su rata gana, entonces obtendrá la misma cantidad que arriesgó, y pasará a aumentar su capital. Si pierde, le será descontada la apuesta. Si se queda en cero, queda fuera del juego.
Una vez que todos los jugadores se hayan quedado sin dinero, el juego comienza otra vez.

```
   1 REM -- .... .    -
   2 REM               -
   3 REM ************
   4 REM * CARRERA   *
   5 REM ************
   6 REM
  10 RAND
  75 CLS
  80 DIM A(4)
  85 DIM B(4)
  90 DIM R(4)
  95 DIM A$(4,8)
 100 PRINT AT 10,2 "CUANTOS JUGADORES SON?(2-4)"
 103 INPUT P
 106 IF P>4 OR P<2 THEN GOTO 105
 110 CLS
 200 FOR N=1 TO P
 204 PRINT
 205 PRINT ,"   JUGADOR Nº   ",N,"NOMBRE?"
 210 INPUT A$(N)
 220 PRINT A$(N)
 230 NEXT N
 235 GOSUB 4000
 250 FOR N=1 TO P
 256 LET R(N)=200
 257 NEXT N
 290 CLS
 295 LET U=0
 300 FOR N=1 TO P
 303 IF R(N)=0 THEN GOTO 350
 305 PRINT ,,,,A$(N) "UD. TIENE $";R(N)
 308 PRINT "CUANTO LE APUESTA AL NRO. ",N,"?"
 310 INPUT B(N)
 312 IF B(N)>R(N) THEN GOTO 310
 314 PRINT "$"; B(N)
 320 NEXT N
 330 GOTO 690
 350 PRINT ,,A$(N) "UD.PERDIO EL DINERO..."
 355 LET U=U+1
 356 IF U=P THEN GOTO 400
 360 NEXT N
 370 GOTO 690
 400 GOSUB 4000
 405 CLS
 410 PRINT AT 8,5 "ESTAN TODOS QUEBRADOS"
 415 PRINT AT 12,0, "JUEGAN OTRA VEZ? S/N"
 420 INPUT I$
 425 IF I$="S" THEN GOTO 435
 426 GOTO 5000
 435 PRINT AT 10,5 "LOS MISMOS JUGADORES? S/N"
 440 INPUT I$
 445 IF I$="S" THEN GOTO 250
 450 GOTO 75
 690 GOSUB 4000
 700 CLS
 705 FOR L=2 TO P+5
 710 PRINT AT L,0,"+"
 720 NEXT L
 750 FOR L=1 TO P+5 STEP 2
 760 PRINT AT L,0 "- - - - - - - - - -"
 770 NEXT L
 780 PRINT AT 0,7 "CARRERA DE RATAS"
 840 FOR N=1 TO P
 845 LET A(N)=20
 850 NEXT N
1000 FOR N=1 TO P
1010 LET L=N+N-3
1020 LET A(N)=A(N)-SGN (RND*3-.5)-(A(N)=1)
1040 GOSUB 2000
1050 NEXT N
1060 GOTO 1000
2000 PRINT AT L,A(N)
2010 PRINT AT L-1,A(N) "  O  "
2020 PRINT AT L+2,A(N) "  --  "
2030 PRINT AT L+3,A(N)
2050 IF A(N)=0 THEN GOTO 2200
2100 RETURN
2200 FOR N=0 TO 4
2205 PRINT AT L,10
2210 FOR Z=0 TO 6
2211 NEXT Z
2225 PRINT AT L,10 "GANADOR"
2230 FOR Z=0 TO 6
2231 NEXT Z
2235 NEXT N
3000 FOR N=1 TO P
3010 IF A(N)=0 THEN GOTO 3050
3020 IF A(N)<>0 THEN GOTO 3030
3030 LET R(N)=R(N)-B(N)
3040 NEXT N
3045 GOTO 290
3050 LET R(N)=R(N)+B(N)
3060 NEXT N
3065 GOTO 290
4000 FOR Z=0 TO 50
4010 NEXT Z
4020 RETURN
5000 PRINT ,,,"OKAY, BYE..."
5100 STOP
5200 SAVE "RATAS"
5300 RUN
```

CALENDARIO 2

COMP: CZ1000/1500 TK83/85
CONF. 16 K
CLAS. PER

Es un programa tan útil como divertido, pues le permite visualizar la hoja calendario en el mes y año que desee, dentro del siglo XX.

PANTALLA

```
CALENDARIO

AÑO ?
1985

DOM LUN MAR MIE JUE VIE SAB
     01  02  03  04  05  06
 07  08  09  10  11  12  13
 14  15  16  17  18  19  20
 21  22  23  24  25  26  27
 28  29  30  31

OTRA FECHA (S N)
```

```
   1 LET X=8
   2 LET Y=7
   3 LET Z=0
   4 LET U=2
  10 PRINT "CALENDARIO",,,,
  15 PRINT    -
  30 PRINT "ANO ?"
  40 INPUT A
  55 LET AI=A
  67 PRINT "MES ?"
  68 INPUT M
  70 PRINT AT 4,0 A,"/" M
  80 IF M>X THEN LET MM=M+6
  90 IF M<=X THEN LET MM=M+10
 100 IF M<=X THEN LET AI=AI-Z
 110 LET C=INT (AI/100)
 120 LET E=AI-100*C
 130 LET E=INT (13*MM-U)/5)+INT (E/4)+INT (C/4)
 140 LET F=E+E+1-4*C
 150 LET F=F-7*INT (F/7)
 160 LET DM=31-(M=4)-(M=6)-(M=9)-(M=11)
 170 IF M<>2 THEN GOTO 210
 180 LET DM=28
 190 IF (A-100*INT (A/100))=U THEN IF (A-400*INT (A/400))=U THEN LET DM=29
 200 IF (A-100*INT (A/100))<>0 THEN IF (A-Y*INT (A/Y))=0 THEN LET DM=29
 210 FOR I=U TO 27
 220 PRINT AT 5,I "▀"
 230 PRINT AT 18,I "▄"
 240 NEXT I
 250 PRINT AT 6,U, "DOM LUN MAR MIE JUE VIE SAB"
 260 FOR I=1 TO DM
 270 LET L=8+INT (F/7)
 280 LET C=4*(F-7*INT (F/7))
 290 IF I<10 THEN PRINT AT L,C "0";I
 300 IF I>=10 THEN PRINT AT L,C,I
 310 LET F=F+Z
 320 IF (F-7*INT (F/7))=U THEN LET F=F+7
 330 NEXT I
 340 PRINT AT 21,2 "OTRA FECHA (S N)?"
 350 INPUT A$
 360 CLS
 370 IF A$="S" THEN GOTO 10
8000 IF M>X THEN LET MM=M-5
```

CAZA FANTASMAS

- COMP. CZ 100/1500 TK83/85
- CONF. 2K
- CLAS. ENT

INSTRUCCIONES:

Este es un programa de persecución en el que hay que evitar encuentros con los fantasmas ("), y comer la mayor cantidad de píldoras de fuerza (*). El jugador sale representado por una C y puede moverse por toda la pantalla, usando las teclas con flechas. El puntaje aumenta cada vez que coma una píldora, y aparecerá recién cuando lo atrape un fantasma.



PANTALLA



```
  1 REM   [illegible]
  2 REM   [illegible]
  3 REM   [illegible]
 10 LET [illegible]=0
 20 LET [illegible]=0
 30 LET [illegible]=0
 40 LET [illegible]=20
 50 LET [illegible]=30
 60 LET [illegible]=INT (RND*10+1)
 70 LET [illegible]=INT (RND*15+1)
 80 LET [illegible]
 90 LET [illegible] AND [illegible] 20 AND [illegible]
    [illegible] AND [illegible] AND [illegible]
 [illegible] LET [illegible] AND [illegible] 30 AND [illegible]
    [illegible] AND [illegible] 5
 [illegible] LET [illegible]
 [illegible] LET [illegible]
 [illegible] PRINT AT A,E;"[illegible]";AT [illegible]
 [illegible] IF [illegible] AND E=F THEN GOTO [illegible]
 [illegible] IF [illegible] AND E=[illegible] THEN GOTO [illegible]
 [illegible]
 [illegible] PRINT [illegible] "PUNTOS";[illegible]
 [illegible]
```

BASKETBALL

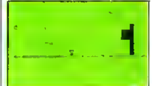
Comp. CZ 1000/1500
Conf. 2 K
Clas: Entretenimiento



INSTRUCCIONES:

Hay que embocar la pelota en el cesto. En pantalla el jugador sale representado por una "A" inversa, que se mueve de izquierda a derecha, presionando cualquier tecla lanzarás la pelota.
El cesto saldrá arbitrariamente en cualquier lado de la pantalla y tu puntaje aparecerá en el vértice superior izquierdo de la pantalla.

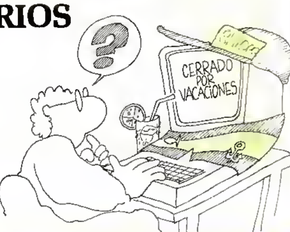
PANTALLA



```
  1 REM   [illegible]
  2 REM   * BASKETBALL *
  3 REM   [illegible]
  4 LET [illegible]
  5 FOR U=SGN PI TO CODE [illegible]
  6 [illegible]
  7 LET U=P1-P1
  9 LET D=INT (RND*CODE [illegible])+CODE [illegible]
 10 FOR G=U TO CODE [illegible]
 11 PRINT AT CODE [illegible],VAL "G+2" [illegible] AT G,D;[illegible]
 12 NEXT G
 13 PRINT AT [illegible] AT P1,PI [illegible]
 20 FOR F=[illegible] TO D-CODE [illegible]
 21 PRINT AT CODE [illegible],F;[illegible]
 22 IF INKEY$<>"" THEN GOTO CODE [illegible]
 30 NEXT F
 40 GOTO [illegible] 102
 50 FOR D=CODE [illegible] TO AI-AI STEP -SGN P1
 57 LET F=F+SGN P1
 60 PRINT AT C,F;[illegible] AT C,F
 63 NEXT D
 80 FOR D=P1-PI TO CODE [illegible]
 81 IF U=A1-PI THEN LET F=F+P1-PI
 82 IF D=CODE [illegible] AND VAL "F+1"=G THEN LET U=[illegible]
 83 IF U=AI/A1 THEN LET S=S+VAL "2"
 84 IF F>=CODE "3" THEN GOTO VAL "102"
 85 PRINT AT D,F;[illegible] AT D,F
 86 NEXT D
104 NEXT U
105 PRINT AT PI-PI,AI-AI;S
108 STOP
109 SAVE "BASK"
110 RUN
```

SUELDOS TEMPORARIOS

Comp CZ1000/1500 TK83/85
Conf 16 K
Clas Comercial

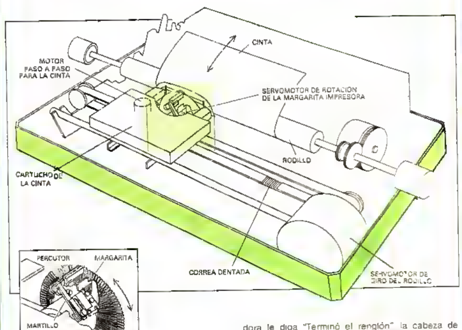


Con este programa se podrá calcular sueldos en forma temporaria. El programa brindará
- Total por horas trabajadas.
- Total de descuentos (jubilación, cuota sindical, etc.)
- Sueldos a cobrar

Al comenzar el programa pedirá
- Pesos por hora.
- Horas trabajadas en total.
- Horas trabajadas en días feriados
- Horas de ausencia por enfermedad.
- Cómo fue el "presentismo".

```
   1 REM  *****=     $4  *******$$
   2 REM  ***********************
   3 REN  $SUELCOS TEMROPPP105*
   4 REM  ***********************
  10 CLS
  11 LET T$=
  12 LET N$=
  13 LET Y$="------------------------
---------
  15 LET E$= ENTRE DPTOS
  18 LET *=0
  16 LET Y=21
  17 LET X$=
  18 LET L=15
  20 GOSUB 800
  22 PSINT YA
  2E OOSUB 600
  30 GOSUB 700
  38 GOTO 872
  50 CLE
  42 GOTO 1000
 E00 CLE
 EA1 SRINT RT 1,0,        N$ A
T 1,17,       $ T$
 502 PETURN
 600 PPINT AT 8,1, HPS,TRABAJADA
S
 610 ARINT AT 11 1  HRS.FERIADO
 620 PRINT AT 13,1, NFS,ENFERMED
AD
 630 RRINT RT 15 1  PRESENTISMO
 640 RRINT RT 17,1  ADELANTO
 650 RETUPN
 700 REN PUTINA DATOS DE ENTARDA
 705 SRINT AT 1,7
 707 PAINT AT Y,A,EA
 710 1NSUT NA
 71E GOSUB 1003
 720 RSINT AT 1 7 N$ AT 1 28  =
 725 PRINT AT Y X,E$
 730 1NAUT T$
 738 GOSUB 1000
 750 RRINT AT 1,28 T$
 75E LET TP=UPL T$
 758 RRINT AT 8 L, "*",PT Y,X E$
 765 1NAUT R$
 768 GOEUB 1000
 768 PRINT RT 9,L,   ,AT 8.29-LE
N A$,RA
 770 LET HT=UAL P$
 778 RRINT RT 11,L "*",PT Y,X E$
 780 1NAUT P$
 78E GOSUB 1000
 790 RRINT RT 11,L    ,AT 11 29-
LEN R$,RA
 798 LET HF=UAL P$
 800 RRINT RT 15.L "*",AT Y,X,E$
 805 1NAUT A$
 810 GOSUB 1000
 81E PRINT AT 13,L,    RT 13,29-
LEN A$ A$
 820 LET HE=VAL P$
 825 PRINT RT 15 L  "*" RT   X
     B1 0)
 830 LET R$=1NKEY$
 832 1F R$= 3  THEN GCTO 840
 833 1F A$, N  THEN GCTO A35
 835 GOTO 830
 83E GOSUB 1000
 836 PPINT AT 15,L,     AT 15 2"
NO
 837 LET P=X
 838 GOTO 800
 840 GOSUB 1000
 845 PRINT AT 15,L      AT 18 2"
51
 848 LET P=1
 850 PRINT PT 17 L  "   AT Y + 
     B1 0)
 8E2 LET A$=1NKE $
 8E3 1F A$= 3  THEN GOTO 8E8
 8E5 1F S$= N  THEN GOTC 8E6
 8EE GOTO 8E1
 8E6 GOSUB 1000
 857 RRINT AT 1",L      A" 17,2"
NO
 858 LET ADE=0
 8E9 GOTO 870
 860 GOSUB 1000
 861 RRINT RT  ,X       MONTO
 862 1NAUT RDE
 863 GDEUB 1000
 865 PRINT AT 17 L      RT 1" 
0 ,AT 17 29-LEN STR$ 1NT ADE AD
E
 870 PRINT AT Y X,"DATOS OK"  B1
 0)
 871 RETURN
 872 LET P$=1NKEY$
 875 1F P$= N  THEN GOTO 880
 876 1F A$= S  TNEN GOTO 40
 878 GOTO 872
 880 CLS
 881 LET N$=
 882 LET T$=
 883 GOTO 20
1000 SRINT RT Y,X,>$
1002 FETUPN
1005 FAST
1009 GOSUB 500
1010 RRINT Y$
1011 LET TH=0
1012 LET LL=15-LEN STR$ 1NT TR
1014 LET TT=1NT (TR*HT*100)/100
1020 RRINT AT 3 1, MPS.TRRB, X
,AT 3,LL,TP,AT 3,28-LEN STR$ 1NT
 TT,TT
1022 LET TN=TT
1024 LET TF=1NT (TA*HF*100)/100
1028 RRINT AT 4 1, MPS.FER,  >
,AT $,LL,TA AT 4,28-LEN STR$ 1NT
 TF TF
1028 LET TH=TH+TF
1030 LET TE=1NT (HE*TR*100  100
1032 RRINT RT 5 1  HRS.ENF.  ,
 AT 5 LL TA AT 5 28-LEN STR$ 1NT
 TE TE
1035 LET TH=TH+TE
1037 1F P=1 THEN LET S=1NT  TH=5
.334) 100
1038 PRINT AT 6 ,  PRESENTISMO
ST 6 28-LEN STR$ 1NT S S
1040 LET TH=TH+S
1050 RRINT  $
1050 PSINT AT 8 1  TOTAL HABERES
  AT 8 28-LEN STR$ INT TH TH
1061 FRINT  $
1064 LET TD=0
1065 LET D=1NT  TH*11  100
1066 ARINT AT 12 1  JUBILACION 1
1  AT 12 28-LEN STR$ 1NT D D
1068 LET TD=
1069 LET D=1NT  TH  100
1070 PSINT AT 11 1  LE  19032
1  AT 11,28-LEN STR$ 1NT D.D
1071 LET TD=TD+D
1074 LET D=1NT  TH*3) 100
1076 PRINT AT 12 1  LE  18010
3  AT 12 28-LEN STRA 1NT D D
1078 LET TD=TD+D
1080 LET D=1NT  TH*2) 100
1082 PRINT RT 13 1  CUOTA SIND,
2  AT 17,28-LEN STR$ 1NT D,D
1085 LET TD=TD+D
1086 PSINT YA
1088 PSINT RT 15 1  TOTAL DEDUC,
   AT 15 28-LEN STR$ 1NT TO,TD
1090 RRINT Y$
1092 RRINT RT 17,1  IMPORTE L1QU
1DO  RT 17,28-LEN STR$ 1NT (TH-T
D),TH-TD
1094 ARINT AT 18 1  ADELANTOS ,A
T 18.28-LEN STR$ 1NT ADE,ADE
1096 SSINT AT 19 1        ,ST
 19 28-LEN STR$ 1NT (TH-TD-ADE)
TH-TD-ADE
1097 SLOW
1098 SRINT RT Y X   CORTE Y LUEG
O PRESIONE
1099 LET R$=1NKEY$
1100 IF CODE A$( 116 THEN GOTO 1
099
1101 LET N$=
1102 LET T$=
1103 GOTO 20
9000 STOP
9000 SAVE "HABERES
9001 CLS
9010 PSINT
9020 FOR J=1 TO 50
9030 NEXT J
9040 RUN
```

La Impresora



Es quizás el periférico mas usado en todo tipo de computadoras. Evidentemente por más exacto y rápido que funcine un computador, si no se pueden representar sus datos de salida en una forma clara y cómoda para su uso humano, de nada servirá. Es por este motivo fundamental que adquieren gran importancia la forma en que estas salidas se imprimen.

Para conocer una impresora vamos a explicar su funcionamiento interno, para despues tratar de hallar una clasificación funcioal de las mismas. Cuando el computador quiera comunicarse con una computadora, le puede mandar los datos en paralelo o en serie (Ver K64 Nº 3 Pag. 42). Existen impresoras que pueden recibir los datos de las dos formas y otras que precisan alguna interface especial pero, sea como sea que lo reciba, la impresora almacena el dato en su memoria interna.

Algunas impresoras tienen una memoria interna grande, donde almecenan miles de bytes para después imprimirlos, pero por lo menos la memoria interna de la impresora es de un renglón, de forma tal que cuando se llene el renglón o cuando la computadora le diga "Terminó el renglón" la cabeza de impresión se mueve y escribe una línea completa.

La forma en que la computadora le dice "Terminó el renglón" es a través de un codigo formado por un byte, interpretado según una tabla llamada de código ASCII.

El código ASCII, es universal y permite comunicar las computadoras de todo el mundo con una gran gama de periféricos y entre sí.

Este mismo código permite saber para cada byte del código qué letra le corresponde, de forma tal que debe haber en la memoria interna una memoria permanente (ROM), que trae grabada la información necesaria para escribir esa letra cuando recibe ese byte.

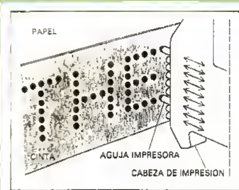
Aquí podemos notar una importante diferencia entre dos tipos de impresoras

a) MATRICIAL O DE AGUJAS
b) MARGARITA O DAISY WHEEL

En el primer caso la información necesaria es la "forma" de la letra y en el segundo, al "lugar" donde se encuentra la letra. Veamos más detalladamente cada uno de estos tipos.

IMPRESORA MATRICIAL O DE AGUJA

Es la impresora más usada en computación, ya que reune características de velocidad y flexibilidad verdadaramente notables



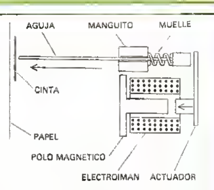


Figura 1

Figura 2

El sistema de impresión por aguja se basa en una cabeza de impresión formada generalmente por siete u ocho bobinas (electroiman) dispuestas en forma circular como se ve en la figura 1.

Cada bobina acciona una chapita (actuador) que actúa como un martillo sobre las agujas. La reposición de las agujas es realizada por la acción de un resorte de compresión, en cada una de ellas. Las agujas si bien están dispuestas en forma circular en la base del cono que forma la cabeza, son guiadas para quedar finalmente formando una sola línea en la punta del cono. Si con un impulso eléctrico se disparan todas las agujas y éstas a su vez pegan en una cinta entintada delante de un papel, quedara impresa una raya vertical formado por 7 u 8 puntos alineados.

El funcionamiento es sumamente sencillo, se trata de dibujar las letras contando solamente con esta línea de agujas. El movimiento de esta línea a través del renglón produce una matríz de puntos. Si las agujas de la cabeza imprimieran siempre, producirían una matriz de puntos que a cierta distancia se vería como una raya gruesa. Por lo tanto para formar una letra lo único que debe hacer la lógica de la impresora es disparar las agujas en el momento correcto. Como se ve en la figura 2, para imprimir la letra "A" primero se disparan las agujas (PIN) 5 6 y 7 simultáneamente y luego solo la 4, y así sucesivamente hasta formar la letra por completo. Evidentemente este sistema permite dibujar cualquier tipo de letra, caracteres o símbolos especiales, siempre que la forma de ese caracter sea almacenada por la memoria ROM que tiene la impresora.

Otra importante ventaja es la velocidad con que imprimen, que como veremos en la próxima nota es mucho más rápida que la impresora tipo margarita.

INGENIERO NINO MORENO

SACANDOLE JUGO A LA CZ y

CZ 1000/1500
TK83/85

El microprocesador Z 80 es el cerebro de nuestras computadoras Sinclair. Sin embargo, no es muy listo por sí solo dado que no es capaz de entender sentencias como PRINT o LET. Cuando usamos estos comandos, no estamos contándole directamente al Z 80 qué es lo que queremos que ejecute, sino que son analizados primero por el INTERPRETE. Este intérprete configura por sí solo un programa del computador y está alojado entre las direcciones 0 y 8191 de la memoria ROM. Este programa está escrito en el llamado "lenguaje máquina" o "código máquina", lenguaje que el Z 80 puede entender directamente. Este lenguaje consiste en números entre 0 y 255. La tabla de estos números y sus mnemónicos las encontraremos en el manual BASIC. Los mnemónicos son etiquetas asignadas a cada código de máquina y describen en forma abreviada cuál es la función de cada código.

Podemos escribir programas en código de máquina, pero no es una tarea fácil.

Las principales ventajas de escribir programas en código máquina radican en que éstos se ejecutan a muy alta velocidad, no toman mucha memoria y son algunas veces más flexibles que el BASIC.

Los tres comandos BASIC que tratan con el código máquina son: PEEK, POKE y USR. PEEK n, nos permite examinar qué código se encuentra almacenado en la dirección de memoria n y POKE m, n coloca el código n en la dirección de memoria m. La rutina USR llama al Z 80 a ejecutar las instrucciones en código máquina almacenadas en una dirección particular.

Antes de usar el comando POKE para colocar números en memoria, debemos reservar el área para ellos. El manual BASIC explica en detalle cómo hacer esto.

Presentamos aquí un listado con cuatro rutinas en código de máquina que imitan funciones que son standard en computadores más costosos, y "agranden" los 8 K ROM agregándole nuevos comandos:



Figura 1.
Sumario de nuevas rutinas

COMANDOS	DESCRIPCION
10 RAND USR IV	Convierte cualquier caracter de pantalla en su inverso.
10 PRINT USR FM	Muestra cuántos bytes libres quedan en memoria.
10 REM 10.2, SIN A	Estas son dos sentencias pseudo-DATA.
500 REM ABE, IKE	Pueden estar en cualquier sitio del programa. Cada entrada pueden ser números o letras seguidos de coma
10 GOTO Z	Genera el listado del programa en inverso.
10 RAND USR RS	Ejecuta el comando RESTORE. Este debe ser usado antes de que cualquier DATA sea leída. Puede usarse para leer una misma data muchas veces.
10 LET A$ = " " 20 RAND USR RD 30 LET A = VAL A$	Hace el READ de la próxima entrada DATA. Omitir la línea 30 si sólo se leerán "sting" de caracteres.



El listado 1, es el programa cargador de los listados de código de maquina de las figuras 2 a 5.

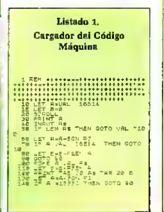
Podemos verificar las direcciones y el checksum mostrado por el programa con las direcciones y checksum de los listados. Si se comete un error, simplemente pulsamos ENTER para borrar la última entrada, y continuar normalmente con la carga.

Los usuarios de 1 K deberán usar CONT para seguir, cuando se produzca la detención del programa por overflow de pantalla.

Luego de tipear el último número, el programa se detendrá.

Tendremos ahora que borrar todas las líneas menos la primera, tipeando para esto el número de línea y luego ENTER.

Con la línea 1 en memoria, tipee y corra (RUN) el listado número 2. Este programa (listado 2) prepara para hacer SAVE a la cinta, por lo tanto debemos preparar el grabador para que recibe el SAVE. Antes de SAVE, tendremos que pulsar ENTER. La pantalla parpadeará un momento y luego comenzará con el SAVE. Ahora el código de máquina está ubicado en la parte superior de memoria, cualquiera sea el tamaño que estemos usando.



Listado 1.
Cargador del Código Máquina

```
   1 REM ++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++
  10 LET A=VAL  16514
  15 LET B=0
  20 SCROLL
  30 PRINT A
  40 INPUT A$
  50 IF LEN A$ THEN GOTO VAL "10
0
  60 LET A=A-SGN PI
  70 IF A VAL  16514  THEN GOTO
10
  80 LET E=E-PEEK A
  90 GOTO 10
 100 POKE A
 110 LET
 120 PRINT TAB 10 A$ TAB 20 B
 130 LET A=A+SGN PI
 140 IF A THEN GOTO 30
```

Listado 2.

```
  10 SAVE  2
  20 LET A=VAL  PEEK 16389+256+P
EEK 16388
  30 INPUT B$
  40 POKE 16515 128
  50 LET A=A-128
  60 LET B=A+3
  70 LET C=VAL  16558
  80 GOSUB VAL  130
  90 LET B=A+29
 100 LET C=VAL  16566
 110 GOSUB 130
 120 PRINT USR 16634
 130 LET E=VAL  INT (B 256)
 140 LET D=B-E*256
 150 POKE C D
 160 POKE C+SGN PI E
 170 RETURN
```

Listado 3

```
   1 SAVE  3
  10 LET AE=VAL "PEEK 16388+256+
PEEK 16389
  25 LET AD=AS+42"
  30 LET FM=AS+79
  40 LET IV=AF+84
  50 REM COMANDOS  USR (AM IV,RS
 AD  AND GOTO Z 9995
  60 PRINT USR FM
  70 RAND USR RS
  75 FOR L=1 TO 3
  80 LET A$=
  90 RAND USR AC
 100 IF L<3 THEN PRINT A$
 105 NEXT L
 110 LET Z=VAL A$
9995 LIST
9999 RAND USR IV
```

Figura 2.
Rutinas Restora y Data

Listado [illegible]		Direcc	Cod	Maq		Checksum
RESTORE	LD HL 16514	16514	33	250	64	347
FIND	LD A, CODE "REM"	16517	62	234		683
	LD BC,65536	16519	1	255	255	1154
	CPIR	16522	237	177		1568
	LD BC,5	16524	1	5	0	1578
	AND A	16527	167			1781
	SBC HL,BC	16528	237	66		2044
	LD A,118	16530	62	118		2224
	CP (HL)	16532	190			2414
	JR Z,BELOW	16533	40	5		2459
	ADD HL,BC	16535	9			2468
STASH	LD (16507),HL	16536	34	123	64	2689
	RET	16539	201			2890
BELOW	ADD HL,BC	16540	9			2899
	JR FIND	16541	24	230		3153
READ CARACTER	LD HL, L(16507)	16543	42	123	64	3382
	LD A ,118	16546	62	118		3562
	CP (HL)	16548	190			3752
	CALL Z,FIND	16549	208	33	64	4053
	LD A, (HL)	16552	126			4179
	INC HL	16553	35			4218
	JR STASH	16554	24	236		4474
READ STRING	LD HL,(E-LINE)	16556	42	20	64	4600
	DEC HL	16559	43			4643
	PUSH HL	16560	229			4872
	DEC HL	16561	43			4915
	DEC HL	16562	43			4958
	EX (SP),HL	16563	227			5185
LOOP	PUSH HL	16564	229			5418
	CALL READ CARACTER	16565	205	159	64	5842
	POP HL	16568	225			6067
	LD B,CODE'.'	16569	6	26		6099
	CP B	16571	188			6283
C.DONE	JR Z,DONE	16572	80	13		6336
	LD(HL),A	16574	119			6455
	INC HL	16575	35			6490
INC LEN$	EX (SP),HL	16576	227			6717
	INC (HL)	16577	52			6769
	JR NZ,NO CARRY	16578	32	3		6804
	INC HL	16580	35			6839
	INC(HL)	16581	52			6891
	DEC HL	16582	43			6934
NO CARRY	EX (SP),HL	16583	227			7161
DO MORE	AND A	16584	167			7328
	JR NC,LOOP	16585	48	233		7609
DONE	EX (SP),HL	16587	227			7836
	POP HL	16588	225			8061
	NOP	16589	0			8061
	JP ROM	16590	195	157	20	8433

SACANDOLE JUGO A LA CZ Y TK

Figura 3.
Rutina de Bytes Libres

	Listado Assembler	Direcc	Cod Maq			Checksum
FREE MEM	LD HL,(STKEND)	16593	42	28	64	8567
	LD B,H	16596	68			8635
	LD C,L	16597	77			8712
	LD HL,0	16598	33	0	0	8745
	ADD HL,SP	16601	57			8802
	CP A	16602	191			8993
	SBC HL,BC	16603	237	66		9296
	LD B,H	16605	68			9364
	LD C,L	16606	77			9441
	RET	16607	201			9642

Figura 4.
Rutina de Inversión

	Listado Assembler	Direcc	Cod Maq			Checksum
IV	LD HL,(D-FILE)	16608	42	12	64	9760
	LD B,H	16611	68			9828
	LD C,L	16612	77			9905
	LD D,59	16613	22	59		9986
	SLA D	16615	203	34		10223
LOOP	LD HL,(VARS)	16617	42	16	64	10345
	LD A,(BC)	16620	10			10355
	CP D	16621	186			10541
	JR Z,AGAIN	16622	40	3		10584
	ADD A,128	16624	198	128		10910
	LD (BC),A	16626	2			10912
AGAIN	INC BC	16627	3			10915
	CP A	16628	191			11106
	ABC HL,BC	16629	237	66		11409
	JR NZ,LOOP	16631	32	240		11681
	RET	16633	201			11882

Aplicaciones prácticas:

El listado 3 demuestra el uso de varios de los comandos. También declara las variables RS, RD, IV, FM y Z. En este listado Z es leida en una linea REM.

Debemos poner a grabar el grabador antes de hacer RUN al listado 3, porque esto pone en SAVE al computador. Luego de esto se deberán borrar todas las líneas excepto la última, de lo contrario el computador se pondrá en SAVE nuevamente. No deben usarse NEW o CLEAR, porque así las variables pueden pasar a la cinta.

El procedimiento para cargar los nuevos comandos es muy sencillo, simplemente LOAD listing 2. A continuación, cualquiera de las rutinas con comandos extendidos.

Precaución. Estos listados funcionarán sólo con la misma configuración de memoria que tenía el computador antes de ser SAVEados a la cinta.

Como algunas aplicaciones, podemos por ejemplo, ver el caso en que se tengan que imprimir muchos títulos con poca memoria, probablemente el computador indique un error "4" indicando un overflow de pantalla. Esto puede ser prevenido con la siguiente linea.

50 IF USR FM • 100 THEN CLS

Esta linea evita el tipear "CONT" para que el programa arranque nuevamente. Pero, la pantalla se borrará tal vez antes de que se pueda leer toda la linea, para prevenir esto, agregaremos unas lineas más:

50 IF USR FM • 100 THEN PAUSE 40000
60 IF USR FM • 100THEN CLS

Esto detendrá el programa dando



Figura 5.
Rutina de Re-ubicación

Listado Assembler		Direcc	Cod Maq			Checksum
Relocate	LD HL,(RAMTOP)	16634	42	4	64	11992
	LD BC,-120	16637	1	136	255	12384
	ADD HL,BC	16640	9			12393
	PUSH HL	16641	229			12622
	PUSH HL	16642	229			12851
	EXX	16643	217			13068
	POP BC	16644	193			13261
	EXX	16645	217			13478
	LD HL,16514	16646	33	130	64	13705
	POP DE	16649	209			13914
	LD BC,120	16650	1	120	0	14035
	LDIR	16653	237	176		14448
	EXX	16655	217			14665
	DEC BC	16656	11			14676
	OUT 253,A	16657	211	253		15140
	JP NEW	16659	195	203	3	15541

tiempo para leer con comodidad, pulsando cualquier tecla,continuará limpiando primero la pantalla. La versión que se elija, dependerá de la aplicación a la que se la destine. Se puede utilizar la rutina de inversión para crear interesantes efectos visuales, o por ejemplo, hacer mas descansada la lectura desde la pantalla etc.

Si hay que recordar números, listas, etc., entonces las rutinas de RESTORE y READ son ideales para esto. Se pueden usar las sentencias REM para guardar importantes números de teléfono, fechas, cumpleaños etc. Pueden ser utiles en inventarios, coordenadas de pantallas para PLOT, direcciones de código de máquina o palabras. El listado 4 es un ejemplo de directorio telefonico por computador, y corre con 1 K.

El listado 5 graficará cualquier funcion que le definamos. Primero debemos entrar en la función en términos de X. Por ejemplo: si buscamos graficar F (x) = (4X + 3X +2) entonces haremos el input con:

4 * X * X + 3 * X + 2

Luego necesitamos fijar los límites de X superior e inferior si plotteamos la curva "sen X66 probablemente elijamos como limites cero (0) y dos pi. Los limites verticales son calculados automáticamente. Almacenaremos en sentencias REM las funciones a graficar junto con sus dos parámetros superior e inferior

Listado 4.

```
  1 SAVE "
  2 LPRINT "LISTADO 4"
  3 LLIST 10
  4 LPRINT "GOTO 0"
  5 STOP
 10 REM WILLIE 020-222256 AR 03-2345878 HUGO 025-245585
 20 PRINT "ENTRE NOMBRE"
 30 INPUT A$
 40 RAND USR RS
 50 LET B$=
 60 RAND USR RD
 70 IF B$="OUT OF DATA" THEN GO TO VAL "20"
 80 LET C$=
 90 RAND USR RD
100 IF B$<>A$ THEN GOTO VAL "50"
110 PRINT C$
120 GOTO VAL "20"
130 REM OUT OF DATA
```

Listado 5.

```
  1 SAVE "5"
  2 LPRINT "LISTADO 5"
  3 LLIST 10
  4 LPRINT "GOTO 0"
  5 STOP
 10 LET F=SGN PI
 20 GOSUB CODE " COPY "
 25 IF F THEN PRINT "FUNCION
 30 INPUT A$
 35 IF F THEN PRINT "LIMITE INF
."
 40 INPUT X
 45 IF F THEN PRINT "LIMITE SUP
."
 50 INPUT A
 60 LET K=X
 70 FAST
 80 IF F THEN CLS
 85 LET F=NOT PI
 90 LET DX=(A-X)/CODE "Z"
100 LET H=VAL A$
110 LET L=H
120 FOR I=NOT PI TO CODE "Z"
130 IF H<VAL A$ THEN LET H=VAL
A$
140 IF L>VAL A$ THEN LET L=VAL
A$
150 LET X=X+DX
160 NEXT I
170 LET X=K
180 SLOW
190 FOR I=NOT PI TO CODE "Z"
200 PLOT I,VAL "43"*(VAL A$-L)/
(H-L)
210 LET X=X+DX
220 NEXT I
230 GOSUB VAL "400"
240 GOTO VAL "30"
300 RAND USR RC
310 FOR A=VAL "12" TO VAL "10"
320 FOR B=NOT PI TO VAL "24" ST
EP VAL "8"
330 LET A$=
340 IF B OR A<NOT PI THEN RAND
USR RD
350 IF NOT B AND A=-SGN PI THEN
LET A$=STR$ A
360 PRINT TAB B,A$
370 NEXT B
380 NEXT A
390 REM *** ELIJA OPCION *** 0
ACION,FN RANGO ,INPUT  SU ,   F
N.
400 INPUT B
420 IF NOT B THEN RETURN
430 RAND USR RS
440 FOR A=SGN PI TO VAL "3*B+B"
450 LET A$=
460 RAND USR RD
470 NEXT A
480 FOR A=-SGN PI TO SGN PI
481 LET B$="
500 RAND USR RD
510 IF A<NOT PI THEN LET A$=B$
520 IF NOT A THEN LET X=VAL B$
530 NEXT A
540 LET A=VAL B$
550 GOTO VAL "50"
1000 REM SIN X 0,2*PI,COS X,0,2*
PI TAN X,-1,1,SIN X+SIN (2*X)+SI
N (3*X) 0 4*PI ASN X -1,1,LN X,.
1 2.7,EXP X,-1 1,ABS (X-INT X-.5
) 0,3,1/X,.1.2 EXP -(X*X),-2.2
```

CONOCIENDO EL BUZON

ING. PEDRO E. COLLA

En general, el T82068 soporta en su conector para cartridge dispositivos exteriores de prácticamente cualquier tipo, pues están allí las principales líneas del bus. Ello no es contradictorio con que primeriamente el mismo este orientado a la utilización de software en ROM (memoria de lectura solamente). La memoria del computador tiene una distribución tal como se especifica en la Figura 1. Nótese que hay una zona en la cual se superponen la memoria ROM principal y la llamada de "Extensión". En realidad no es que ambos bancos puedan funcionar o estar visibles al mismo tiempo, sino que se realiza la conmutación entre ambos en la medida que se quiere ejecutar lo que contiene uno u el otro. En una anterior entrega en esta publicación se dio un ejemplo de tal situación.



Independientemente del contenido original, la memoria se divide en secciones cuyo tamaño es de 8 kbytes denominadas "cuartiles" o "chunks", teniendo el hardware la posibilidad de eliminar mediante instrucciones de software adecuadas una o más de esas zonas. Esto facilita enormemente el agregado externo de software en reemplazo del interno pues es posible anular este último mediante el apagado del "chunk" correspondiente y por supuesto su reemplazo con el que nos resulte conveniente.

De esta manera es posible colocar en el buzón software en ROM que reemplace parte o la totalidad de ROM del computador, sin tener que recurrir a engorrosos artificios circuitales, como era necesario hacer por ejemplo en el computador TS1000.

En realidad, las cosas son más fáciles aun pues el computador como parte de su rutina de encendido (cuando la pantalla se vuelve negra) realiza por sí mismo una lectura en el conector de cartridge, y si detecta que allí hay algo, toma los recaudos para utilizarlo y que el resto del computador no interfiera con ello.

Se prevee que pueden existir dos tipos de software en cartridge, lo que se denomina LROS y su contrapartida el AROS.

El primero de ellos (LROS) está pensando para software en lenguaje de máquina que reemplace al (los) ROM original del computador para proveer al mismo de una modalidad de funcionamiento enteramente diferente. Claros ejemplos de uso de esta alternativa serian hacer que el principal lenguaje fuera FORTH y no BASIC, o aprovechando la existencia del procesador Z80 el lograr que se descarte el modo de funcionamiento del Sinclair para transformarlo en un sistema con diskettes bajo CP/M.

En cambio en el caso del AROS se prevee la utilización de software que necesite del ROM Sinclair, es decir de aplicaciones o juegos. Además en el caso del AROS el contenido del cartridge puede estar tanto en lenguaje de máquina como en BASIC y por cierto no elimine al ROM principal.

Físicamente un cartridge es una plaqueta de circuito impreso conteniendo uno o más chips de memoria ROM o EPROM. El tipo de memoria a utilizar está fundamentalmente dado por el volúmen de producción que se prevea. En el caso de software comercial, las memorias tipo ROM son posiblemente la solución de menor costo mientras que para proyectos esencialmente unitarios, es infinitamente más practica la utilización de memorias EPROM (memorias de

Una de las características que distinguen al computador Timex-Sinclair 2068 de su similar europao, el SPECTRUM, es al conector para cartridga, más vulgarmente conocido como buzón. Es posible agregarlo a esta máquina, paro en forme opcional.

Lectura Solamente y Borrables). Las primeras posiciones de la misma contienen la información necesaria para que al realizar el Proceso de inicialización el computador pueda conocer si el contenido de la misma es de tipo LROS o AROS y cuales son los chunks de memoria que resultan afectados.

En el caso del LROS, que es el que nos interesa, son los primeros 5 bytes los que contienen tal información y su estructura es la siguiente

0000	No es utilizada
0001	Contiene el valor 01 indicando LROS.
0002/0003	Dirección de primera instrucción a ejecutar
0004	Especificación de chunks a anular.

Nótese que si no hay cartridge conectado el computador, "ve" solamente direcciones conteniendo el valor 255 y por lo tanto al leer la posición 0001 se da por notificado que no hay ningún LROS para ejecutar (y tampoco ningún AROS cuyo valor para posición 0001 es 02) y continúa operando normalmente.

Una vez que el computador se notifica que existe una sección de LROS, utiliza el quinto byte para saber que secciones de memoria debe reemplazar con el. Las posiciones segunda y tercera le dan la información acerca de a que dirección de memoria se deberá transferir la ejecución del programa luego de haber finalizado la inicialización.

FIGURA 1
MAPA DE MEMORIA DEL COMPUTADOR TS2068

DIRECCIONES	CONTENIDO		CHUNK
00000 A 08191	ROM	ROM	CHUNK 0
08192 A 16383	PRINCIPAL	EXTEND.	CHUNK 1
16384 A 24575			CHUNK 2
24576 A 32767			CHUNK 3
32768 A 40959			CHUNK 4
40960 A 49151			CHUNK 5
49152 A 57313	RAM		CHUNK 6
57314 A 65535	PRINCIPAL		CHUNK 7

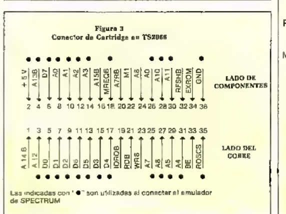


Figura 3
Conector de Cartridge en TS2068

Las indicadas con "●" son utilizadas al conectar el emulador de SPECTRUM

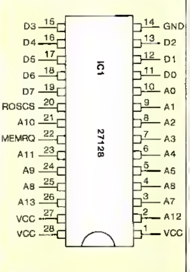


Figura 2
Conexiones del EPROM al conector del Cartridge

CONVERSION DE LA TS 2068

La especificación de que chunks son anulados y cuáles no, se logra numerando cada uno de 0 a 7 (8 chunks de 8 kbytes cada uno suman 64 Kbytes, la totalidad de memoria), cada bit de esta posición refiere a un chunk específico, siendo el bit 0 el que controla el segmento que va entre las direcciones 0 y 8191, el bit 2 al que transcurre entre 8192 y 16383 y así sucesivamente. De tal manera que si el bit asociado tiene un valor "1", el chunk correspondiente no debe ser anulado, mientras que si su valor es "0" se debe proceder a la desconexión del mismo. Para clarificar esto con un ejemplo supongamos que queremos reemplazar con el contenido de un cartridge la totalidad del ROM del computador, el que vimos en Fig. 1 que se extiende entre las posiciones 0 y 16383. Los primeros cinco bytes del cartridge deberán entonces contener

0000 000
0001 001
0002 000
0003 019
0004 252 1111 1100 anula los chunks 0 y 1.

Es interesante apuntar que la mayoría del software ideado para computadores SPECTRUM no es compatible para su ejecución en máquinas TS2068, y por lo tanto aprovechando la similitud del hardware entre ambas es factible hacer que el computdor TS2068 "emule" un SPECTRUM colocando el ROM de este último en reemplazo del original del primero.

Con este cambio el TS2068 sin duda pierde potencia, pues no puede utilizar en forma sencilla los joysticks, el sintetizador musical y algunas instrucciones especiales del BASIC (ON ERR, STICK, RESET, SOUND, etc.) Pero se gana en el acceso al constantemente renovado software para SPECTRUM que en calidad y cantidad supera el disponible para máquinas TS2068. Para lograr ello deberemos reproducir en una EPROM al contenido del ROM de un computador SPECTRUM y modificar las posiciones de memoria necesarias para cumplir los requisitos de un LROS.

Tendremos entonces que concurrir a una de las numerosas casas especializadas en grabación de EPROM portando el chip "vacío" junto con una ROM de cualquier computador SPECTRUM, posiblemente prestada por un amigo, para que se obtenga un "duplicado" de la misma. Normalmente con un bajo o nulo cargo adicional es posible que antes de asentar la copia se realicen las moficaciones que se consignan en la presente, para que dicho chip pueda funcionar como LROS.

En las Figuras 2 y 3 se brindan el esquema de conexiones del circuito integrado y el diagrama del zocalo de cartridge del computador. Nótese que debido a la extrema sencillez del conexionado, no se justifica al armado de una plaqueta de circuito impreso, siendo posible el armado del circuito con técnicas de soldado punto a punto o "wire-wrapping".

Como integrado EPROM se utiliza un 27128 cuya capacidad es de 16K bytes X 8 bits, por lo que la totalidad del ROM SPECTRUM puede ser albergada en un sólo integrado. En caso de dificultades en la obtención del citado elemento electrónico, nada impide lograr el mismo propósito mediante dos chips 2764 (8 Kbytes X 8 bits) o cuatro chips 2732 (4Kbytes X 8 bits) de más fácil obtención, aunque en ambos casos se complicará ostensiblemente el aspecto mecánico del montaje.

Para que el integrado grabado funcione correctamente, debe tenerse en cuenta que las primeras posiciones no podrán coincidir exactamente con el ROM SPECTRUM, pues deben contener los datos necesarios para que durante la inicialización se disponga de la información para utilizarlo correctamente. Para ello se aprovecha un hueco de algunos bytes en la posición 19 del ROM SPECTRUM para transferir allí las instrucciones ubicadas originalmente en las primeras direcciones. Por ello se debe grabar en forma diferente las primeras 30 posiciones sobre el integrado de EPROM, siendo el contenido final de las mismas el expresado por la Figura 4. El resto de las posiciones son un fiel reflejo de las contenidas en el ROM SPECTRUM.

Al encender el computador con el cartridge ocupado por el circuito que nos ocupa, y si hemos hecho bien las cosas, el computador debería "inicializarse" como lo hace normalmente, llegando a exhibir durante algo menos de un segundo la leyenda de "copyright" normal para a continuación volver a inicializarse para aparecer al final de dicho proceso el mensaje de "copyright dal computador SPECTRUM" de allí en adelante a todos los efectos prácticos del computador dejará de comportarse como un TS2068 u será un SPECTRUM. Si ésto no ocurre significará que se ha cometido algún error en la grabación de la EPROM o en el conexionado del circuito para lo cual habrá que revisar cuidadosamente ambos puntos. En ningún caso se puede conectar o desconectar el cartridge con el computador encendido pues el mismo se puede dañar. Exceptuando esta última precuación, es improbable que un error en el circuito produzca daño alguno y mucho menos permanente.

FIGURA 4
POSICIONES INICIALES PARA EMULADOR DE SPECTRUM

DIRECCION	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09
00000	255	001	019	000	252	255	255	255	042	093
00010	092	034	095	092	024	067	195	242	021	017
00020	255	255	024	019	042	093	092	126	205	125
00030 A 16383	IGUAL A ROM SPECTRUM									



2do CONCURSO TRIMESTRAL

PATROCINA

SANWA



1er. Premio: Una Consola COMMODORE 64
2do. Premio: Una Consola SPECTRUM
3er. Premio: Dos Pasajes a Bariloche
4to. Premio: Un Grabador para Computadora
5to. Premio: Una Impresora Alpha Com 32

Condiciones para participar en el certamen:

1: Los programas deberán ser originales e inéditos, pudiendo cubrir todas las áreas: educativos y de cálculo, uso comercial, entretenimiento, personal y utilitarios para programación, hasta 64 K. 2: El criterio de elección se basará en: originalidad de la idea, método de programación, efectos gráficos y/o sonoros, documentación, presentación y ahorro de memoria. 3: Se enviarán a K-64 grabados en un cassette y acompañados por el listado correspondiente con pantallas y explicación sobre la utilidad y manejo del programa. 4: Puede remitirse mas de un programa por cassette, en lo posible grabados dos veces, para mayor seguridad. 5: El concurso K-64 se realizará trimestralmente. El cierre de la recepción de los trabajos será el 15/9/85. 6: K-64 se reserva el derecho de publicación de los programas enviados (como asimismo de la devolución del material recibido).

El cassette deberá ser enviado con su caja y con los datos del programa y del autor, como así también de la computadora para la cual está destinado.

SELECCION MENSUAL

Mensualmente se seleccionarán 50 Programas, los que se harán acreedores a los siguientes premios: Calculadoras, Máquinas de fotos, cassettes con programas, cassettes vírgenes, Becas para Cursos, etc.

Los Programas seleccionados continúan en Concurso para la gran final Trimestral.

Presentando este cupón obtendrán un 10% de descuento de las compras que realicen en SANWA S.A. y un 50% para los cursos en EPI.

RETIRO DE CUPONES: K-64: Carrito 1320 · SANWA: Av. Corrientes 5198, Florida 683 · EPI: Suipacha 946 · 1er. piso, Viamonte 1470 · 8º Piso "B", Florida 683, Av. Corrientes 2198 · Radio del Plata: Av. Santa Fe 2043.

AUSPICIAN





EDITOR DE TEXTOS

TI 99/4A



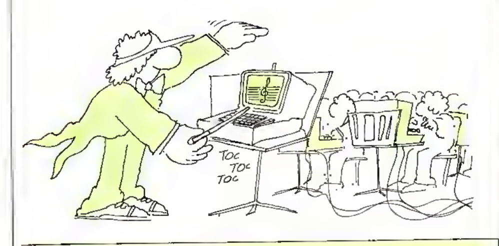

```
2 CALL CLEAR :: FOR A=1 TO 10 :: CALL COLOR(A,6,1):: NEXT A :: CALL SCREEN(16):
: CALL COLOR(1,6,16):: CALL CHAR(33,"0000000F0F0F0F0"):: CALL CHAR(34,"F0F0F0FD
")
3 CALL CHAR(35,"FFFFFFFFF0F0F0F0"):: CALL CHAR(36,"F0F0F0F0F0F0F0F0"):: CALL CHA
R(37,"FFFFFFFFFFFFFFFF"):: CALL CHAR(38,"FF00FF00FF00FF00")
4 CALL CHAR(40,"1000F000F000F000"):: CALL SCHR(1,3,37,2):: CALL SCHR(4,6,37,2)
:: CALL VCHAR(4,1,31,2):: CALL HCHAR(5,1,33):: CALL HCHAR(1,1,34)
5 CALL HCHAR(1,3,35):: CALL SCHR(1,A,34):: CALL HCHAR(3,5,33):: CALL HCHAR(3,6,
35):: CALL HCHAR(3,1,36):: CALL HCHAR(1,7,34)
6 CALL HCHAR(1,8,33):: CALL SCHR(1,A,36):: CALL HCHAR(5,12,33):: CALL HCHAR(2,1
,34):: CALL HCHAR(2,1,38,30):: CALL VCHAR(1,38,40,1)
7 A$="EMPRESA PARA INFORMATICA" :: B=8 :: C=3 :: GOSUB 10 :: A$="PROGRAMAS PARA
EL" :: B=11 :: C=3 :: GOSUB 10 :: A$="(CROCOMPUTADOR TI-99/4A" :: B=11 :: C=1 :
: GOSUB 10
8 A$="FABRICADO Y DISTRIBUIDO EN" :: B=10 :: C=3 :: GOSUB 10 :: A$="LA REPUBLIC
A ARGENTINA POR" :: B=21 :: C=3 :: GOSUB 10 :: B$=""GDI"(NTELIGENCIA ARGENTINE"
 :: B=11 :: C=3
9 GOSUB 10 :: FOR D=1 TO 700 :: NEXT D :: GOTO 11
10 FOR D=1 TO LEN(A$):: B$=SEG$(A$,D,1):: E=ASC(B$):: CALL HCHAR(B,C+D,E):: NE
XT D :: RETURN
11 ON WARNING NEXT :: CALL CHARSET :: CALL CLEAR
12 CALL SCREEN(11):: DISPLAY AT(11,2):"   EDITOR DE TEXTOS" :: FOR TT=1 TO 500
:: NEXT TT
13 DISPLAY AT(1,2)BEEP ERASE ALL:"ESTE PROGRAMA LE PERMITIRA  OPERAR EN PANTALLA
 A POR IMPRESORA CON UN VERDADERO EDITOR DE TEXTOS "
14 DISPLAY AT(1,1):"LOS TEXTOS PUEDEN QUEDAR    ALMACENADOS EN CASSETTE O EN DISK
ETTE. "
15 DISPLAY AT(11,2):"NO REDUNDAREMOS EN MAYORES  EXPLICACIONES, PUESTO QUE   PEN
SAMOS QUE EL PROGRAMA ES AUTOEXPLICATIVO EN SUS APLICA-"
16 DISPLAY AT(16,2):"CIONES Y MODOS DE OPERACION,SOLO ESPERAMOS QUE DISFRUTE DE
ESTE UTILITARIO REALIZADO PARA SU TI-99/4A."
18 FOR EE=1 TO 200 :: NEXT EE
20 DISPLAY AT(11,1):"OPRIMA LA TECLA ENTER" :: ACCEPT AT(24,24)BEEP:ANS$
21 CALL CLEAR
81 OPTION BASE 1 :: DIM C$(700),D$(31)
111 DATA NUL,SOH,STX,ETX,EOT,ENQ,ACK,BEL,BS,HT,LF,VT,FF,CR,SO,SI,DLE,DC1,DC2,DC3
,DC4
311 DATA NAK,SYN,ETB,CAN,EM,SUB,ESC,FS,GS,RS,US
111 FOR I=1 TO 31 :: READ D$(I):: NEXT I :: E$=CHR$(1)&CHR$(12)&CHR$(27)&CHR$(1
8)&CHR$(11)&CHR$(72)&CHR$(13)&CHR$(27)&"S" :: F$="1" :: G=80 :: CALL CLEAR :: CA
LL SCREEN(11)
331 I$="LO LAMENTO,MEMORIA LLENA "
361 CALL CLEAR
371 DISPLAY AT(3,1):"      EDITOR DE TEXTOS         Seleccione comando" :: DI
SPLAY AT(1,1):"1-Borrar  2-Agregar 3-Reemplazar  linea    linea    linea "
391 DISPLAY AT(10,1):"4-Borrar  5-Crear   6-Grabar     All:   titulo   titulo" ::
 DISPLAY AT(13,1):"7-Listar  8-Cargar  9-Reemplazar  titulo   titulo   palabra"
411 DISPLAY AT(16,1):"0-Para ver los comandos"
421 DISPLAY AT(20,1):"CUANTOS CARACTERES POR LINEA QUIERE  ?"|G :: ACCEPT AT(21,1
)VALIDATE(DIGIT)SIZE(-3):D :: IF (G<1)+(D>254)THEN 421
451 PRINT :: ON ERROR 3941 :: CALL SCREEN(14):: INPUT "Accion ? ":D
111 IF N=-1 THEN 3901 :: N=N+1
481 IF (N<1)+(N>9)THEN 491 ELSE 511
491 CALL CLEAR :: DISPLAY AT(1,1):"((ERROR DE SELECCION))" :: GOTO 371
511 ON D GOSUB 561,541,711,891,1031,1101,1251,1131,1841,2141
521 IF D=9 THEN CALL CLEAR :: GOTO 361
531 GOTO 451
541 REM
561 IF I>0 THEN 601
581 PRINT :"     NO HAY TITULO   ": :: GOTO 681
601 GOSUB 2971 :: IF J=0 THEN 681
621 I=I-1 :: FOR F=J TO I :: C$(F)=C$(F+1):: NEXT F :: C$(I+1)="" :: PRINT :"Bor
rada,titulo renumerado":
681 RETURN
691 REM
711 CALL SCREEN(10):: IF I>0 THEN 751
731 PRINT :"   NO HAY TITULO   ": :: GOTO 871
751 PRINT :"Agregar despues" :: GOSUB 2971 :: IF J=0 THEN 871
101 PRINT :"Entre nueva linea-" :: H$=STR$(I):: GOSUB 2031 :: I=I+1 :: FOR F=I T
O J+1 STEP -1 :: C$(F)=C$(F-1):: NEXT F :: C$(J)=H$ :: PRINT :"Linea agregada,ti
tulo renumerado": :
871 RETURN
881 REM
891 IF I>0 THEN 941
921 PRINT :"   NO HAY TITULO   ": :: GOTO 1021
941 PRINT :"Reemplace" :: GOSUB 2971 :: IF J=0 THEN 1021
971 PRINT :"Entre linea de reemplazo-": :: H$=STR$(J):: GOSUB 2031 :: C$(J)=H$ ::
 PRINT :"Linea reemplazada": :
1021 RETURN
1031 REM
1041 FOR F=1 TO I :: C$(F)="" :: NEXT F :: I=0 :: PRINT : :"  AREA DE MEMORIA LI
MPIA": : :: RETURN
1101 REM
1121 CALL SCREEN(10):: CALL CLEAR :: PRINT : : :"--- Liste para tipee ---
entre ^^^ para fin": :
1111 H$=STR$(I+1):: GOSUB 2031 :: IF LEN(H$)<>4 THEN 1211
1201 IF SEG$(H$,1,3)="^^^" THEN 1241
1211 I=I+1 :: C$(I)=H$ :: GOTO 1111
1241 RETURN
1251 REM
1271 IF I>0 THEN 1311
1291 PRINT : :"   NO HAY TITULO   ": : :: GOTO 1401
1311 PRINT : :"Entre rangos titulo a grabar": : :: GOSUB 3101 :: PRINT
1341 INPUT "Grabe en                 1=cassette  2=diskette":K :: IF K<>1 T
HEN 1381
1361 GOSUB 3311 :: GOTO 1401
1381 IF K<>3 THEN 1111
```


K64

```
1391 GOSUB 3991 :: GOTO 1111
1111 PRINT :: CALL SOUND(500,220,1,659,1):: GOTO 1341
1441 RETURN
1451 REM
1411 PRINT :"Entre el rango de lineas de los titulos a imprimir": :: GOSUB 3101
 :: PRINT : : : :: INPUT "Salida por IMPRESORA o por PANTALLA ? (I/P)":J$ :: PR
INT : : :
1531 IF J$="I" OR J$="P" THEN 1601
1541 FOR I=L TO J :: R=F :: GOSUB 1681 :: PRINT :[illegible] :: NEXT I :: GOTO 1851
1601 CALL CLEAR :: DISPLAY AT(12,1):"Ingrese los ajustes p/ impresor": : :: DISPLAY
 AT(16,1):"DA=" :: ACCEPT AT(16,4)BEEP:I$ :: CALL CLEAR :: PRINT : : :" imprimi
endo el fondo  ": : : :: [illegible]
1611 OPEN #1:"RS232.BA=9.DA="&I$,VARIABLE 132 :: PRINT #1:E$ :: PRINT #1
1672 FOR F=L TO J :: R=F :: GOSUB 1681 :: PRINT #1:SEG$(X$,1,LEN(X$)-1):: NEXT F
 :: PRINT #1:CHR$(12):: CLOSE #1
1851 RETURN
1861 REM
1881 IF I=0 THEN 2021
1901 CALL SOUND(500,220,1,659,1):: PRINT : : :"EL TITULO YA TIENE LOS DATOS":
1921 PRINT "Entre ""C"" p/ CLEAR [illegible]   entre ""M"" p/ CARGAR otro" :: INPUT @
$ :: IF [illegible]="C" OR [illegible]=""THEN 1901
1951 GOSUB 1931 :: FOR D=1 TO 50 :: NEXT D :: GOTO 861
1981 IF [illegible]="B" OR [illegible]=""THEN 2021
1991 CALL SOUND(500,220,1,659,1):: PRINT :: GOTO 1921
2021 PRINT
2011 INPUT "CARGA/MERGE desde        1=cassette 2=diskette ?":X :: IF X<1 [illegible]
[illegible] 2071
2051 GOSUB 3301 :: GOTO 2131
2071 IF X<>2 THEN 2101
2081 GOSUB 3401 :: GOTO 2111
2101 CALL SOUND(500,220,1,659,1):: PRINT :: GOTO 2031
2111 RETURN
2141 REM
2161 PRINT :"Palabra a reemplazar" :: [illegible] :: GOSUB 2931 :: [illegible]
[illegible] :: P=LEN([illegible]):: IF P>0 THEN 2251
2231 PRINT :"NO PUEDO REEMPLAZAR UNA PALABRA NULA": : :: RETURN
2251 Q=0 :: PRINT :"Entre palabra de reemplazo" :: GOSUB 2931 :: [illegible]
[illegible] :: PRINT :"Entre rango de busqueda." :: GOSUB 3101 :: PRINT : :" [illegible]
SCANDO [illegible]": : :: FOR I=L TO J :: IF LEN([illegible])<P THEN 2441
2341 S=POS(C$(I),[illegible],1):: IF S=0 THEN 2441
2361 [illegible]
[illegible] Q=Q+1 :: PRINT "Linea cambiada":[illegible] :: GOSUB 2461 :: PRINT [illegible]
[illegible] NEXT I :: PRINT : :[illegible] :: RETURN
2111 PRINT :"Entre rango de lineas a imprimir" :: GOSUB 3101 :: PRINT : :"
 IMPRIMIENDO TEXTO   ": : :: [illegible]
2192 OPEN #1:"RS232.BA=9.DA="&[illegible],VARIABLE 132 :: PRINT #1:E$ :: FOR F=L
 TO J :: PRINT #1:C$(F):: NEXT F :: CLOSE #1 :: RETURN
2441 REM
2711 [illegible] :: FOR B=1 TO [illegible] :: [illegible] :: IF V<33
THEN 2771
2731 IF V<128 THEN 2801
2701 V=V-128
2771 [illegible]
2811 NEXT B :: RETURN
2821 REM
2851 [illegible] :: IF LEN([illegible])>[illegible] THEN 2881 ELSE 2951
2881 PRINT :"UNA LINEA DEMASIADO LARGA [illegible]": : :: CALL SOUND(500,120,1,651,1):: FOR I=
[illegible] TO 1 STEP -1 :: IF SEG$([illegible],I,1)=" " THEN 2931
2921 NEXT I
2931 [illegible]:: PRINT [illegible]
2951 [illegible] :: RETURN
2971 REM
2991 PRINT
3001 INPUT "Numero de linea ":J :: IF [illegible] THEN 3011
3011 PRINT : :"     NO EXISTE LA LINEA    ": : :: GOTO 3001
3011 IF J=0 THEN 3091
3031 PRINT :"Linea anterior":J;"[illegible]": : :: R=J :: GOSUB 3701 :: PRINT X$
3091 RETURN
3111 REM
3111 PRINT "La ultima linea en el listado":[illegible] :: PRINT :"Entre el numero de la
primer linea," :: INPUT "(Entre cero p/todo el listado)":L :: IF L>0 THEN 3101
3161 I=1 :: J=1 :: GOTO 3291
3101 IF L>0 [illegible] THEN 3221
3201 PRINT :"   NUMERO FUERA DE RANGO  ": : :: GOTO 3121
3221 PRINT :: INPUT "Entre el numero de la ultima linea":J :: IF J>L THEN 3271
3251 J=L :: GOTO 3291
3271 IF J<I THEN 3291
3281 J=I
3291 RETURN
3301 REM
3321 OPEN #1:"CS1",INTERNAL,INPUT ,FIXED 192 :: INPUT #1:AA :: BA=1 :: FOR I=BA
 TO AA+BA :: INPUT #1:C$(I):: I=I+1 :: NEXT F :: CLOSE #1 :: RETURN
3391 REM
3411 OPEN #1:"CS1",INTERNAL,OUTPUT,FIXED 192 :: PRINT #1:[illegible] :: FOR I=L TO
J :: PRINT #1:C$(I):: NEXT F :: CLOSE #1 :: RETURN
3481 REM
3561 GOSUB 3491 :: OPEN #1:"DSK"&[illegible],INPUT ,SEQUENTIAL,INTERNAL,VARI
ABLE 254 :: INPUT #1:AA :: BA=1 :: FOR F=BA TO AA+BA :: INPUT #1:C$(F):: [illegible]
 :: NEXT F :: CLOSE #1 :: RETURN
3591 REM
3611 GOSUB 3491 :: OPEN #1:"DSK"&[illegible],OUTPUT,SEQUENTIAL,INTERNAL,VARI
ABLE 254 :: PRINT #1:[illegible] :: FOR I=L TO J :: PRINT #1:C$(I):: NEXT I :: CLOS
E #1 :: RETURN
3691 REM
3711 PRINT
3721 INPUT "Nro. drive 1,2,3 ?":CA :: IF (CA<1)+(CA>3)=0 THEN 3771
3741 PRINT :: CALL SOUND(500,220,1,659,1):: GOTO 3771
3771 PRINT : :"ENGANCHE TECLA ALPHA LOCK  Entre el nombre del titulo." :: INPUT
 "(no mas de 10 caracteres)  ":G$ :: IF [illegible] THEN 3831
3801 PRINT :: CALL SOUND(500,220,1,659,1):: GOTO 3771
3831 RETURN
3841 ON ERROR 3921 :: CALL ERR(DA,EA)
3871 CALL SOUND(500,110,1,220,1,659,1):: IF [illegible] THEN 3891 ELSE 1101
3891 C$(I)="" :: C$(I-1)="" :: C$(I-2)="" :: C$(I-3)="" :: C$(I-4)="" :: I=I-5
 : PRINT : [illegible] :: GOTO 1941
3901 IF DA>[illegible] AND DA<[illegible] THEN PRINT : :"LO LAMENTO,ERROR I/O " :: GOTO 3911
3911 PRINT : :"<<< ERROR ### >>>" :: GOTO 1111
3921 CALL ERR(DA,EA,FA,GA):: PRINT "ERROR";DA;"EN LINEA";GA
3931 ON ERROR 3941
3951 CLOSE #1 :: GOTO 3971
[illegible] CALL ERR(DA,EA)
3971 RETURN 451
3981 CALL CLEAR
```

MUSICA EN LA TI 99

TI 99/4A




```
2 CALL CLEAR :: FOR I=1 TO 11 :: CALL COLOR(I,7,2):: NEXT A :: CALL SCREEN(1)::
CALL COLOR(1,1,1):: CALL CHAR(33,"00000000FOFOFOF"):: CALL CHAR(1,"FOFOFOFO")
: CALL CHAR(35,"111FFFFFFFOFOFOF"):: CALL CHAR(36,"FOFOFOFOFOFOFOF"):: CALL CHA
R(31,"111FFFFFFFFFFFFF"):: CALL CHAR(38,"F1111100FF00FF00")
: CALL CHAR(40,"F000F000F000F000"):: CALL VCHAR(1,3,37,2):: CALL HCHAR(1,6,37,21
:: CALL VCHAR(4,1,37,1):: CALL HCHAR(3,1,33):: CALL HCHAR(1,1,34)
: CALL HCHAR(3,5,35):: CALL HCHAR(1,1,31):: CALL HCHAR(3,5,33):: CALL HCHAR(5,1,
35):: CALL HCHAR(1,1,36):: CALL HCHAR(4,7,34)
: CALL HCHAR(1,1,33):: CALL HCHAR(1,1,36):: CALL HCHAR(5,11,33):: CALL HCHAR(2,1
,1):: CALL HCHAR(1,1,31,30):: CALL VCHAR(1,30,40,1)
7 A$="EMPRESA PARA INFORMATICA" :: B=8 :: C=3 :: GOSUB 10 :: A$="PROGRAMAS PARA
EL" :: B=11 :: C=3 :: GOSUB 11 :: A$="MICROCOMPUTADOR TI-99/4A" :: B=11 :: C=1 :
: GOSUB 11
8 A$="FABRICADO Y DISTRIBUIDO EN" :: B=18 :: C=5 :: GOSUB 11 :: A$="LA REPUBLIC
A ARGENTINA POR" :: B=20 :: C=1 :: GOSUB 11 :: A$=""SOT"INTELIGENCIA ARGENTINA"
:: B=22 :: C=3
: GOSUB 11 :: FOR D=1 TO 100 :: NEXT D :: GOTO 11
11 FOR D=1 TO LEN(A$):: B$=SEG$(A$,D,1):: E=ASC(B$):: CALL HCHAR(B,C-1+D,E):: NE
XT D :: RETURN
11 ON WARNING NEXT :: CALL CHARSET :: CALL CLEAR
120 CALL CLEAR :: I=77 :: CALL SCREEN(4):: FOR G=1 TO 11 :: CALL COLOR(G,11,1)::
 NEXT G :: FOR I=96 TO 143 :: READ C$ :: CALL CHAR(N,C$):: NEXT N :: DISPLAY AT
(1,1):CHR$(11)
150 DISPLAY AT(1,1):CHR$(97)&CHR$(99)&CHR$(99):: DISPLAY AT(1,12):CHR$(97)&"
"&CHR$(11):: CALL HCHAR(1,15,99,1)
170 DISPLAY AT(1,1):CHR$(97)&"     "&CHR$(99):: CALL HCHAR(1,11,11,5):: DISPLAY
AT(1,1):CHR$(97)&"     "&CHR$(99):: CALL HCHAR(1,31,99,1)
190 DISPLAY AT(1,1):CHR$(1)&"     "&CHR$(99):: CALL HCHAR(10,12,99,9):: FO
R I=11 TO 15 :: CALL HCHAR(I,11,99,1):: NEXT I
200 CALL HCHAR(11,11,99,1):: FOR N=100 TO 106 :: CALL HCHAR(14,N-87,1):: NEXT N
:: DISPLAY AT(1,1):CHR$(99)&CHR$(99)&CHR$(107)&CHR$(108)&CHR$(109)&CHR$(11)&C
HR$(111)" "&CHR$(113)&CHR$(99)&CHR$(11)
220 CALL HCHAR(11,11,99,1):: FOR N=111 TO 121 :: CALL HCHAR(16,N-102,1):: NEXT
N :: CALL HCHAR(17,13,99,1):: FOR N=122 TO 121 :: CALL HCHAR(17,N-110,1):: NEXT
N
240 DISPLAY AT(18,1):CHR$(99)&CHR$(122)&CHR$(130)&CHR$(112)&CHR$(115)&CHR$(132)&
CHR$(133)&CHR$(112)&CHR$(131)&CHR$(99)&CHR$(11)
250 CALL HCHAR(18,1,112,32):: FOR N=135 TO 111 :: CALL HCHAR(18,N-122,N):: NEXT
N
111 DISPLAY AT(21,1):"MUSICA EN LA TI-99/4A      ********************      OP
RIMA UNA TECLA.........."
270 F=1 :: CALL SPRITE(#1,113,11,11,F,-37,0,#2,103,11,16,F,-21,0):: READ J ::
IF J=800 THEN F=77 :: RESTORE 141 :: GOTO 270
280 CALL KEY(1,K,L):: IF L<>0 THEN 500
290 FOR N=1 TO 5 :: CALL SOUND(-900,1,115,312,885,187.1,30,-4,385):: NEXT N :: C
ALL DELSPRITE(ALL):: GOTO 271
500 CALL CLEAR :: CALL DELSPRITE(ALL):: DISPLAY AT(1,7):"PARA HACER MUSICA EN LA
 TI-99/4A,SOLO SIGA LAS INS- TRUCCIONES.LAS TECLAS SON  LOS NUMEROS 10-91."
510 DISPLAY AT(8,1):"ESTAS TECLAS SE CORRESPONDEN CON LAS DE UN INSTRUMENTO." ::
DISPLAY AT(1,1):"VEAMOSLOS":"  1= DO":"  2= RE":"  3= MI"
330 DISPLAY AT(1,1):"  4= FA":"  5= SOL":"  7= LA":"  1= SI":"  8= DO":"  9= RE
":"  0= MI" :: DISPLAY AT(24,1):"OPRIMA CUALQUIER TECLA"
350 CALL KEY(0,K,L):: IF L=0 THEN 350
360 CALL SOUND(100,800,0):: CALL SOUND(100,1320,1)
371 M=-211 :: D=2^(1/12):: CALL CLEAR :: DISPLAY AT(10,1):"OPRIMA:
        1) PARA VER MENU          1) ORGANO                2) SOUND
ICA"
390 DISPLAY AT(14,1):"         3) CLAVICORDIO              1) BOOGIE
     1) 111"
400 CALL KEY(0,K,L):: IF L=0 THEN 400
111 CALL SOUND(99,800,1):: CALL SOUND(99,1320,1):: IF K=11 THEN 100
130 IF K=82 THEN 371
110 IF K<111 OR K>52 THEN 400
130 ON K-10 GOTO 460,111,471,1000
460 CALL CLEAR :: DISPLAY AT(1,10):"ORGANO" :: DISPLAY AT(21,1):"OPRIMA "1" PARA
 VER EL MENU"
470 DISPLAY AT(1,1):"  1 2 3 4 5 6 1 1 1 1":"   * * * * * * * * * *":"   S 0 C
 I E 1 1 1 1 0 C" :: DISPLAY AT(23,1):"OPRIMA "0" PARA OIR TEMA." :: P=0 :: GOSUB
 1330
490 CALL KEY(1,K,L):: IF L=0 THEN 490
500 IF K<170 THEN 551
510 CALL SOUND(200,800,1):: CALL SOUND(100,1320,3):: P=P+1 :: IF P=10 THEN P=0 :
: GOSUB 1331 :: GOTO 190
521 ON K GOSUB 1340,1350,1270,1300,1400,1120,1440,1160,1100,3271
530 IF K=82 THEN 110
111 IF K<111 OR K>157 THEN 190
130 ON K-17 GOTO 660,560,170,580,590,111,620,630,190,650
560 CALL SOUND(1,220,0,440,1):: GOTO 190
570 CALL SOUND(1,117,1,194,9):: GOTO 490
580 CALL SOUND(1,212,1,511,1):: GOTO 490
590 CALL SOUND(1,111,1,580,1):: GOTO 490
600 FOR W=1 TO 1999 :: NEXT W :: CALL SCREEN(1)
111 CALL SOUND(1,130,1,661,1):: GOTO 190
620 CALL SOUND(1,111,0,670,1):: GOTO 111
130 CALL SOUND(1,192,1,781,1):: GOTO 190
640 CALL SOUND(1,190,0,881,1):: GOTO 190
650 CALL SOUND(1,111,1,990,1):: GOTO 190
660 CALL SOUND(1,523,1,1146,1):: GOTO 190
170 CALL CLEAR :: DISPLAY AT(1,1):"ARMONICA" :: DISPLAY AT(24,1):"OPRIMA "0" PA
RA VER EL MENU"
680 DISPLAY AT(1,1):"  1 2 5 1 1 1 1 1 1 1":"   * * * * * * * * * *":"   1 0 C
 1 1 1 5 0 0 C" :: DISPLAY AT(23,1):"OPRIMA "1" PARA OIR TEMA." :: P=0 :: GOSUB
 1330
100 CALL KEY(0,K,L):: IF L=0 THEN 700
111 IF K=82 THEN 371
720 IF K<170 THEN 750
730 CALL SOUND(500,800,1):: CALL SOUND(100,1320,1):: P=P+1 :: IF P=11 THEN P=1 :
: GOSUB 1330 :: GOTO 100
```

```
111 ON P GOSUB 1340,1350,1371,1380,1100,1120,1440,1440,1400,32717
150 IF I<48 OR I>57 THEN 700
160 ON I-47 GOTO 860,771,180,790,800,810,820,830,840,850
170 CALL SOUND(N,220,1,262,9):: GOTO 700
180 CALL SOUND(N,247,0,294,9):: GOTO 700
190 CALL SOUND(N,262,1,330,1):: GOTO 700
800 CALL SOUND(N,294,0,349,9):: GOTO 700
810 CALL SOUND(N,330,1,392,9):: GOTO 700
820 CALL SOUND(N,349,1,440,9):: GOTO 700
830 CALL SOUND(N,392,0,494,9):: GOTO 700
840 CALL SOUND(N,440,0,523,1):: GOTO 700
850 CALL SOUND(N,494,1,587,1):: GOTO 700
860 CALL SOUND(N,523,1,659,9):: GOTO 700
870 CALL CLEAR :: DISPLAY AT(3,1):"CLAVICORDIO" :: DISPLAY AT(24,1):"OPRIMA '0'
PARA VER EL MENU"
880 DISPLAY AT(1,1):"  1 2 3 4 5 6 7 8 9 0";"  ' ' ' ' ' ' ' ' ' ' ' ";"  A B C
D E F G H I C" :: DISPLAY AT(23,1):"OPRIMA '1' PARA OTRO TEMA." :: P=0 :: GOSUB
1330
900 CALL KEY(0,I,L):: IF L=0 THEN 900
910 IF I<>48 THEN 940
920 CALL SOUND(300,800,1):: CALL SOUND(100,1320,1):: P=P+1 :: IF P=10 THEN P=0 :
: GOSUB 1330 :: GOTO 900
930 ON P GOSUB 1340,1350,1370,1380,1100,1420,1440,1460,1480,32761
940 IF K=82 THEN 370
950 IF I<48 OR I>57 THEN 900
960 ON I-47 GOTO 1060,970,980,990,1000,1010,1020,1030,1040,1050
970 R=220 :: I=440 :: GOTO 1070
980 R=247 :: I=494 :: GOTO 1070
990 R=262 :: K=523 :: GOTO 1070
1000 R=294 :: I=587 :: GOTO 1070
1010 R=330 :: I=659 :: GOTO 1070
1020 R=349 :: I=698 :: GOTO 1070
1030 R=392 :: I=784 :: GOTO 1070
1040 R=440 :: I=880 :: GOTO 1070
1050 R=494 :: I=988 :: GOTO 1070
1060 R=523 :: I=1046 :: GOTO 1070
1070 FOR F=0 TO 30 STEP 7 :: CALL SOUND(1,R,F,I,F):: NEXT F :: GOTO 900
1080 CALL CLEAR :: DISPLAY AT(3,1):"BOOGIE" :: DISPLAY AT(20,1):"OPRIMA '0' PAR
A VER EL MENU"
1090 DISPLAY AT(10,1):"  1 2 3 4 5 6 7 8 9 0";"  ' ' ' ' ' ' ' ' ' ' ' ";"  A B C
D E F G H I C"
1100 DISPLAY AT(12,1):"BOOGIE BOOGIE...";"ES:    3 5 3 1 3            1
  6   1   0";"OPRIMALAS DE 1 EN 1...."
1110 CALL KEY(0,I,L):: IF L=0 THEN 1110
1120 IF I=82 THEN 370
1130 IF I<48 OR I>57 THEN 1110
1140 ON K-47 GOTO 1240,1150,1160,1170,1180,1190,1200,1210,1220,1230
1150 R=220 :: GOTO 1250
1160 R=247 :: GOTO 1250
1170 R=262 :: GOTO 1250
1180 R=294 :: GOTO 1250
1190 R=330 :: GOTO 1250
1200 R=349 :: GOTO 1250
1210 R=392 :: GOTO 1250
1220 R=440 :: GOTO 1250
1230 R=494 :: GOTO 1250
1240 R=523 :: GOTO 1250
1250 CALL SOUND(-1/2,1,1,1,1):: CALL SOUND(-1/1,R,1210*1,1,R*0,0):: CALL SOUND(
R,R*0*1,1,.R*1*1,1):: CALL SOUND(1,R*1*1,1,R*0*1,1)
1260 CALL SOUND(1,R*0*1,1,.5*R*0*1,1):: CALL SOUND(R,R*0*7,0,R*0*7,0)
1270 CALL SOUND(1,R*0*9,1,.R*0*9,0):: CALL SOUND(R,R*0*1,1,R*0*1,1)
1280 CALL SOUND(1,R*0*10,0,.R*0*10,1):: CALL SOUND(R,R*0*10,0,R*0*11,1)
1290 CALL SOUND(R,R*0*9,0,.R*0*9,1):: CALL SOUND(1,2*R*0*1,0,R*0*9,1)
1300 CALL SOUND(R,R*0*1,1,.R*0*1,1):: CALL SOUND(1,R*0*1,0,R*0*1,1)
1310 CALL SOUND(1,R*0*1,0,.R*0*1,1):: CALL SOUND(1,2*R*0*1,1,R*0*5,0):: GOTO 1
110
1320 ON I GOSUB 1340,1350,1370,1380,1400,1420,1440,1460,1480,32711
1330 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"MARI TIENE UN CORDERITO..";"E
S:    111 1000 111 577                543 4555 544 543"" :: RETURN
1340 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"BOGA BOGA EL BOTERO.....";"ES
:      33340 54540                    0077  5553  765432" :: RETURN
1350 CALL HCHAR(11,1,32,320):: DISPLAY AT(12,1):"TITILA,TITILA, ESTRELLITA..";
"ES:  3377 007                    6655 443"
1360 DISPLAY AT(16,1):"      7766 554                      7766 554
      3377 001                    6655 443" :: RETURN
1370 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"PUENTE DE LONDRES";"ES:  1111
 541 154 541                7871 111 1153" :: RETURN
1380 CALL HCHAR(10,1,32,320):: DISPLAY AT(11,1):"CORRIENDO ES EL CAMPO.";"ES:
 1757 875 54 54              7757 175 4565 63 3"
1390 DISPLAY AT(11,1):"      33570 00001                   1757 875 1565 63 3" :: R
ETURN
1400 CALL HCHAR(11,1,32,020):: DISPLAY AT(10,1):"ESTE VIEJO HOMBRE.";"ES: 757
757 8763 456 1173 333 14567 744 65" :: RETURN
1420 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"VIEJO MACDONALD...";"ES:    7
77 4501 1100H               4 777 4531 17301"
1430 DISPLAY AT(16,1):"      1 777 777 1771 777            4 777 1001 99007" :: RET
URN
1440 CALL HCHAR(12,1,32,320):: DISPLAY AT(12,1):"CAMPANAS DE NAVIDAD.";"ES:  5
55 555 57311                666 6655 5544 54 7"
1450 DISPLAY AT(16,1):"      555 555 11345                 661 6655 577 1 1 3" :: R
ETURN
1460 CALL HCHAR(12,1,32,120):: DISPLAY AT(12,1):"TRES CIEGUITOS.....";"ES:  54
1 541 765 765               7009 89077"
1470 DISPLAY AT(16,1):"      1000 89077                    7009 89077
   1 5 1 1" :: RETURN
1480 CALL HCHAR(11,1,32,020):: DISPLAY AT(12,1):"HACE MUCHO TIEMPO.";"ES:  0 0
45 561 075                  1 654 1 511"
1490 DISPLAY AT(16,1):"      3 345 561 870                 7 654 50 0" :: RETURN
1500 DATA 000000000AAAAAEE,04BADADEIEIEAEEE,4040AOE0E4EAEAEE,E1111EEEEEE11111,EE
EEEEEEEEEEEFEE,EEIEEFEOF0C00,EF1,FF,110F,EEEEEE1E0E0201
1520 DATA EEEEEEEEEEEEEEEE,1C10F0F1E110E0C,000000001100000,90010F1F33A00A0A,00F
0FEFEFE0001,0000E0F0C0AC0007,FFFFFFFFFFFFFFFF,2E1E0E0E0400[illegible]
1540 DATA [illegible]
00FFFFFF500FF,A0C7FFFFFF2F00FF
1560 DATA 0000000FFFF5200FF,0201010101C1C1E,EFEFEFEFEFEFEFEF,0F1F3530F3030F3F,FFF
F2000FF000FFF,FFFF5200FF00FFFF,FFFFA500FF00FFFF
1570 DATA FFFF2A00FF00FFFF
1590 DATA FFFF5200FF00FFFF,F0F0AC00FC004FCFC,3F3F3F3F3F3F3F3F,FFFFAF5EA05AA55,FF
55AA55AA55AA55AA,FFFF0F57AA55AAF5,FCFCFCFCFCFCFCFC
1610 DATA 3F3F0F3F0F0F0F0F,F1FFFFFFCFEFCFCFC0,AA55FFAAAAAA2222,A055FFAAAAAA202,F2F
3CF4002430,FFFFFFFF1F1F3F1F
1630 DATA F0F0F0F0F0F0F0F0,FF00FF00FF00FF,40E0E0E04
1640 DATA 110,123,131,147,165,175,196,220,247,262,294,330,349,392,440,494,523,58
7,659,698,784,880,999
```

INTERFACE DE GRABADOR

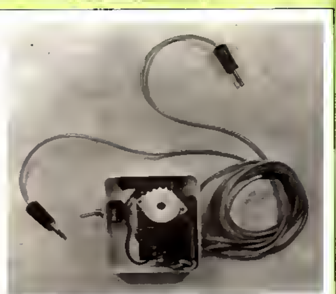
Días pasados me compré una C64, pero lamenlablemente no me elcanzó pera el datassette, por lo que me puse a ver cómo podía conecter la máquina a un grabador común, para lo que tuve que desarmerle el datessette a mi amigo Sergio ante le desazón de su pequeño véslago, que temle no poder disfruter más de guerras intergalácticas ni conducir fórmules uno.

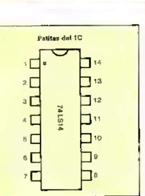
Como fruto de verios experimentos les olrezco este circuito' (que también se puede adquinr como kit). Reemplaze sin problemas el costoso detasetteque, a peser de ser de beja calidad, cueste lo que un super walkmen con em-fm, reloj y entena eléctrice

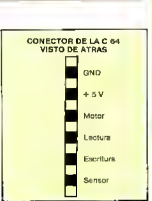
El circuito usa el mismo integrado TTL que el detassete un 74LS14 que es un schmitt-trigger séxtuple que transforma las ondes de sonido haciéndolas blen cuadradites, para que le máquine interprete los unos y ceros. En el circuilo de lectura se puso une lleve, porque algunos grabadores Invierten en la selide la poleridad de la señal; por lo que con le lleve seleccionamos si la leemos tal cual, o la invertimos

En le primera etapa se pusieron doe resistencias para sumarle una tensión contínue e le señal y ubicarle en le zone de disparo de 74LS14 También se dispuso en forma opcionel une bocina piezo-eléctrice de alta impedancia que sirve pare escucher lo que lee la máquine (la bocina se encuenlre en verias casas de electrónica del centro).

El circuito de grabación tiene un preset para ajuatar el nivel de grabación por las dudas sea muy bejo o allo para nuestro grabador (En la primere prueba ponerlo al medio). Como elemenlo opcional se agregó un relé de 6 voltios pere controler el remoto. No se puede en general usar un tranelstor aquí porque el remoto en muchos grabadores corta entre positivo y el borne del motor y si tuviéramos mesa común produciríemos un cortocircuito en le fuente del grebador. El montaje puede hacerse en une plequetite de experimentación y meteria en una cajite de plástico.







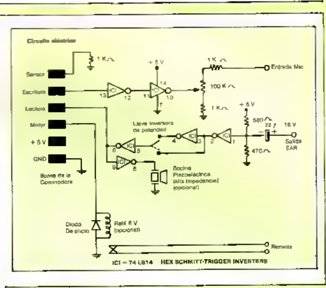
AJUSTE

Con la máquine epagade enchufamos el artetacto (luego de haber eupervisado las conexiones), escri-

PARA COMMODORE

bimos un pequeño programita y tratamos de salvarlo (save "nombre"). Si el grabador permite monitorear por la salida de audífono, se escuchará le grabación en la bocina. Si no se escucha debe estar muy bajo el volumen. Luego rebobinamos y tratamos de leerlo; por la bocinita se escuchara el piii...crunch-crunch. Si no escuchamos nada, levantemos el volumen del grabador. Si tampoco se escucha levantar el volumen de grabación con el preset y volver a grabar.

Una vez que ya ascuchamos el ruido, ponemos la cinta al principio y le damos load "". Si no reconoce el programa, volver de nuevo la cinta y probar con la llave para el otro lado. Cada grabador requiere una posición particular de la llave de lectura



NOTAS

Si cargamos programas con turbo y no le pusimos el relé, una vez que le dimos el LOAD apretemos continuamente una tecla hasta que lo encuentra, porque si lo encuentra y no lo aceptemos enseguida se pasa de largo.

En un próximo artículo veremos la posibilidad de conectar impresoras comunes a la interfase serie de la C64 y todo otro pedido o idea que me acerquen

Hasta la próxima.

Marcelo D. Martinez



EL SISTEMA OPERATIVO DEL COMMODORE 1541



El Commodore 64, como comentáramos en su presentación (K 64 Nº 2, pag 28), es un ordenador de aceptables prestaciones en el nivel de gestión administrativa. En este artículo, efectuaremos una revisión de las utilidades que se pueden obtener empleando unidades de mini discos Commodore 1541 y, en particular del funcionamiento de su programa controlador o "Sistema Operativo".

Luego de meses de utilizar el lentísimo grabador a cassettes ("Quelonio", como algunos lo han bautizado), probablemente habrá empezado a considerar muy seriamente la posibilidad de adquirir una unidad de discos. Es un hecho que el tiempo de carga de programas desde el datassette resulta francamente exasperante y, al mismo tiempo, su utilización nos impide sacar provecho de avanzados sistemas de bases de datos disponibles en el mercado (Superbase 64, The Manager, etc.) por la reducida capacidad de manejo de archivos que tiene este periférico.

Ahora bien, efectuada la adquisición de su nuevo periférico y una buena provisión de disquettes, Ud se encontrará con un manual del usuario notablemente escueto y caracterizado por una buena landa de errores (de hecho, la mayor parte de los manuales de periféricos CBM son muy breves). Dejando de lado los capítulos sobre instalación y encendido, consideramos que la descripción de los co-

mandos del Sistema Operativo de la Disquettera (D.O.S. o Disk Operating System) es muy reducida y la carencia de ejemplos extensos para cada comando desanima al usuario a emplearlos.

LOS COMANDOS DEL D.O.S.:

Veamos a continuación los comandos disponibles en el DOS Versión 2 de las disquetteras Commodore 1541.

OPEN (ABRIR):

Excepto al efectuar la carga, grabación o verificación de un programa, todo operación hacia la disquetera debe iniciarse con esta orden, que dice al D.O.S. que abra un archivo y se prepare a leer o grabar datos o a recibir información sobre el estado del periférico u órdenes especiales. Su sintaxis es
OPEN NA,ND,NC [,"TEXTO"]
Donde NA es el número de archivo dentro de la computadora (también podemos llamarle "vía"). ND es el número de dispositivo, es decir el número que identifica a la unidad de discos; NC es el numero del canal dentro de la disquettera (también podemos llamarle "dirección secundaria") y "TEXTO" es una sarta de caracteres que se emplea o no según el tipo de apertura de archivos que se efectúe.
NA puede valer cualquier número de 1 a 255; ND puede valer de 8 a 11 si bien, por defecto, es 8. Si tuviera necesidad de cambiar dicho valor vea el Capítulo 9 de su manual del usuario para modificarlo apropiadamente. Recomendamos efectuar la modificación por el método de software y no por hardware, ya que la apertura y manipulación del contenido de la disquettera, sin los conocimientos necesarios, pueden traer resultados terribles.
NC, como ya dijimos, es el número de canal por el cual pasará la información dentro de la disquettera. En la 1541 existen 16 de estos canales, pero tres de ellos tienen funciones especiales: los Nº 0 y 1 están reservados para operaciones LOAD, SAVE y VERIFY, y el Nº 15 es conocido como "Canal de Comando" ya que él es el que sirve para dar órdenes específicas al drive y para que éste nos informe el resultado de esas órdenes.
La última parte de la sentencia OPEN, el "TEXTO", varía respecto de la apertura que se realice, veamos a continuación una tabla de comandos OPEN:

SENTENCIA	COMENTARIOS
OPEN 15,8,15	Abre el canal de comando del dispositivo Nº 8. A continuación, se podrá dar una orden al D.O.S. por medio de una sentencia PRINT# que veremos más adelante; o bien leer el estado del dispositivo (lectura de condiciones de error).
OPEN 2,8,2"SEC,S,R"	Abre el archivo Nº 2 hacia el canal 2 del drive número 8. Allí opera sobre el archivo llamado "SEC", que es de tipo S (secuencial) y que se va a leer (la R indica lectura)
OPEN 3,8,5"K64,S,W"	Abre el archivo número tres hacia el canal número 5 del dispositivo 8. Allí escribirá sobre el archivo secuencial K64 (la última W denota escritura).
OPEN 2,8,2"$"	Abre el archivo número 2 el canal número 2 del dispositivo 8. Allí opera sobre el directorio del disco, que se denomina "$". Mientras OPEN está en vigencia es muy recomendable NO efectuar una orden PRINT#2 ya que ésta destruirá parte del directorio.
OPEN 8,8,8,"#"	Abre el archivo numero 8 hacia el canal del mismo número en el drive 8. En este caso, el texto "#" indica que se abre un archivo de tipo "Random" (de acceso aleatorio), que se caracteriza por permitir la lectura de un bloque completo del disco (256 bytes)
A$="REL L"+CHR$(12)	Asigna a la variable A$ el nombre de un archivo relativo más la sarta ",L," y el CHR$(12) que indica la longitud de cada registro de este archivo
OPEN 7,8,3,A$	Abre el archivo 7 el canal 3 del dispositivo 8. El texto A$ le indica que se esta creando un archivo relativo en el cual cada registro tendrá una longitud de 12 bytes
TX$=CDO$+":"+ARCH$	Asigna a la variable TX$ un comando (CDO$), los dos puntos y el nombre de un archivo (ARCH$)
OPEN 15,8,15,TX$	Abre el archivo 15 al canal de comando de la disquettera 8 y le ordena ejecute el comando CDO$ sobre el archivo ARCH$

En la sintaxis que describimos más arriba, y como demostramos en estos ejemplos, todos los componentes de una sentencia OPEN (NA, ND, NC y "TEXTO") pueden ser variables. Al usar estos parametros con variables se debe tener la precaución de asegurar la asignación de valores correctos a las mismas. De lo contrario, pueden generarse condiciones de error tales como la de "DEVICE NOT PRESENT ERROR" (error de dispositivo ausente). Por otro lado, es muy recomendable utilizar el mismo número de archivo y de canal; para de este modo tener clara la relación entre ambos. Así, por ejemplo, el archivo 15 siempre estará relacionado al canal de error de la disquettera

CLOSE (CERRAR):

Concluidas las operaciones con una disquettera (excepto LOAD, SAVE y VERIFY) es imperativa la ejecución del comando CLOSE que cerrará el/los archivos sobre los cuales se estuvo operando. Su sintaxis es muy simple

CLOSE NA

Donde NA es el número del archivo que se desea cerrar. NA puede ser una constante o una variable; y aquí también valen las consideraciones hechas sobre la sentencia OPEN. Cerrar un archivo que nunca se abrió dará por resultado un mensaje "FILE NOT OPEN ERROR" (error de archivo no abierto)

SENTENCIAS DE ENTRADA/SALIDA:

Cuando se trabaja con archivos en

discos, la entrada/salida de datos desde las disquetteras es idéntica a la proveniente del teclado y se efectua por medio de órdenes INPUT#NA, GET#NA y PRINT#NA, siendo NA el número del archivo desde/hacia el cual se efectúan estas entradas/salidas. Obviamente, el archivo NA deberá haber sido abierto antes de efectuar la primera de cualquiera de estas operaciones.

La sentencia INPUT# lee bytes hasta encontrar un CHR$(13) (retorno de carro) y asigna los bytes leidos anteriormente a la variable que se le indique. La sentencia GET#, en cambio, lee un byte y lo asigna a la variable especificada sin discriminar sus contenidos; salvo en el caso en que se intente leer un byte que define un caracter alfabético y cargarlo en una variable numérica. La orden PRINT# imprimirá en el archivo especificado el contenido de la variable seguido de un CHR$(13) o no, según como se cierre la sentencia.

Cuando se lee un archivo del cual se desconoce el formato de sus datos, se recomienda utilizar la sentencia GET# y una variable alfanumérica y analizar uno por uno los bytes leidos desde el archivo para determinar el contenido real del archivo y descartar apropiadamente los separadores de registros. A continuación, veamos algunos ejemplos de instrucciones de entrada/salida:

NEW (NUEVO):

Este comando se utiliza para formatear o reformatear un disco. Según la forma en que se emplee, borrará todo el contenido del disquette ya que "dibujará" sobre la capa magnética del mismo los bloques en los que posteriormente grabará sus programas y datos. Para ordenar este comando utilizamos las sentencias OPEN y CLOSE de la siguiente forma:

OPEN 15,8,15,"N:"+NDS+","+ID $ CLOSE15

Donde NDS, al nombre del disquette, es una sarta de hasta 16 caracteres de longitud e ID$, que representa al identificador del mismo, es otra de 2 caracteres de largo. Dicho comando formateará el disquette identificándolo con el nombre contenido en NDS y el identificador descripto en IDS. Si el disquette ya estaba formateado toda la información que éste contenía se perderá definitivamente.

En ciertas oportunidades, Ud. deseará re-formatear un disco que ya no llena mayor utilidad pero que estaba formateado. Para ganar tiempo en este proceso, el comando de formateo a ordenar será:

OPEN 15,8,15,"N:"+NDS CLOSE15

Una vez más, NDS representa el nombre del disco y puede tener hasta 16 caracteres. El formateo de un disco de acuerdo al último procedimiento aquí descrito no destruye la información contenida en él; con la única excepción del directorio ($) que sí es efectivamente borrado. Por esta razón, se puede recuperar la información de ciertos archivos antiguos si el disco reformateado no se ha usado demasiado.

Una recomendación personal que hacemos a los lectores es la de utilizar nombres de discos normalizados, es decir, denominados de una forma estandarizada. Por ejemplo, todos nuestros discos siguen la convención de tener como nombre un número de cuatro dígitos rellenado con ceros a la izquierda y como identificador el número de disco representado en dos dígitos hexadecimales. De esta manera, el disco más antiguo lleva el nombre 0001 y el identificador 01. Algunos dirán, y qué pasa cuando lleguen al disco número 255 (FF en hexadecimal)? Lo más probable es que, cuando lleguemos a dicha unidad, el número 0001 ya estará fuera de servicio por su excesivo uso. De este modo el ciclo se repetirá.

Otra idea sería utilizar la fecha de formateo inicial como nombre del disco. Así, un disquette formateado el 30 de junio de 1985 podría llevar el nombre "85-06-30" o alguna combinación similar. En este caso, aparte de un nombre siempre distinto para cada disco, tendremos información sobre cuanto tiempo ha estado el mismo en servicio: información que nos permitirá determinar el momento propicio para su reemplazo.

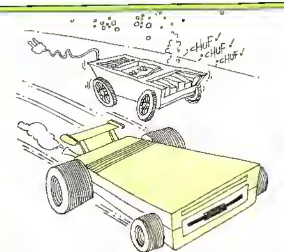
COPY (COPIAR):

Este es un comando que dentro del mismo disco, tiene la capacidad de copiar un archivo e, inclusive, de encadenar hasta cuatro archivos en uno. En el primer caso, su utilidad es reducida ya que no tiene mayor sentido contar con dos versiones del mismo archivo en el mismo disco. Por el contrario, la segunda opción es muy beneficiosa ya que nos permite encadenar diversos archivos y generar un archivo principal.

Supongamos que tenemos cuatro archivos secuenciales que contienen datos referidos a los clientes de cuatro sucursales distintas de una misma empresa de computación, y que la gerencia de ésta desea un informe completo sobre toda su clientela. Una forma de preparar dicho informe sería tratar por separado cada uno de los archivos de las sucursales. Otra solución más económica desde el punto de vista del programador sería encadenar esos cuatro archivos en uno

Lo que sigue es de Ejemplos de ENTRADA/SALIDA:

SENTENCIA	COMENTARIOS
GET#5,A$	Toma un byte del archivo Nº 4 y lo asigna a la variable sarta A$.
GET#3,A	Toma un byte del archivo Nº 3 y su valor real lo asigna a la variable A.
GET#7,A$,B$,A,D,D%	Lee cinco bytes y los asigna según su tipo a las variables especificadas.
INPUT#2,A$	Lee una serie de bytes y los concatena en A$ hasta hallar un CHR$(13). Si esto no sucede hasta el byte número 80, el programa BASIC se interrumpe con un mensaje "STRING TOO LONG" (sarta de caracteres demasiado larga).
INPUT#2,A	Lee una serie de bytes hasta hallar un CHR$(13) o un valor incorrecto para una variable real.
PRINT#3,F,F$	Imprimirá en el archivo 3 la variable A incluyendo un espacio a la izquierda para su signo seguida de un CHR$(13) e, inmediatamente, el contenido de la variable alfanumérica F$



provisono y luego procesar este último.

De esta manera se reduciría el potencial dado que cualquiera de los cuatro archivos separados podría sufrir. Al mismo tiempo, el programador tendría menos que codificar ya que al principio del programa uniría los archivos y luego se dedicaría a procesar el principal.

Para copiar un solo archivo en otro la sentencia es
OPEN 15,8,15,"C:"+NA$+"="+"VA$"
Donde NA$ contiene el nombre de la copia y VA$ el nombre del original.

En el caso de encadenamiento múltiple, el comando será:
OPEN 15,8,15,"C:"+NA$+"="+VA$(1)+","+VA$(2)+","+VA$(3)+","+VA$(4):CLOSE15
Donde NA$ contiene el nombre del archivo de destino y el arreglo VA$ los archivos originales a encadenar.

RENAME (RENOMBRAR):

Este comando permite cambiar el nombre de un archivo existente en el directorio del disco. Para efectuar este cambio solo se debe ingresar.
OPEN 15,8,15,"R:"+NN$+"="+VN$:CLOSE15
Donde NN$ contiene el nuevo nombre que le queremos asignar al archivo y VN$ su nombre actual.

SCRATCH ("RASCAR"):

Este comando, al que nosotros llamamos "KAPUTT", tiene como función la "destrucción" de un archivo. En realidad, la ejecución de este comando no destruye el archivo ni su referencia en el directorio. Lo que si hace es marcarlo como "nulo" o "borrado" (deleted) en su referencia del directorio y hacer que los bloques que ocupe queden disponibles para otro archivo. Esto implica que un archivo que ha sido borrado con este comando por error, se puede recuperar, por supuesto si el disco no fue usado demasiado luego del borrado. Su sintaxis es.
OPEN 15,8,15,"S:"+NA$:CLOSE15
Donde la variable NA$ contiene el nombre del archivo al que se aplicará este comando.

De acuerdo a experiencias de otros usuarios, nos hemos enterado que en ciertas ocasiones el comando SAVE & REPLACE (vea su manual para la sintaxis apropiada) no funciona correctamente, e inclusive ha llegado a destruir algunos archivos. Para evitar el potencial efecto nocivo de dicho comando, nosotros utilizamos una breve rutina que cumple la funcion del SAVE & REPLACE con absoluta confiabilidad. En nuestros programas usamos esta rutina, ubicándola en las primeras líneas, para asegurarnos que la última versión del programa en desarrollo haya sido grabada al empezar su ejecución. De este modo, si durante el programa se produce una interrupción inesperada, de la cual no podemos recuperarnos, aún tendremos el programa en el disco.

Como alternativa, esta rutine se puede colocar en alguna parte del programa que habitualmente no se ejecute; tal el caso de una subrutina que nunca se llama desde el programa o de varias lineas al final del mismo precedidas por une sentencia STOP. De esta manera, desde el modo directo se podrá operar esta rutina sólo las veces que se desee; con el consiguiente ahorro de tiempo que esto implica. Las tres posibilidades aqui comentadas se muestran en el listado Nº 2

INITIALIZE (INICIALIZAR):

Este comando es a la disquettera 1541 lo que la SYS 64738 es a la Commodore 64; es decir, coloca a la máquina en la condición que ésta tiene al momento de ser encendida. En los drives se utiliza cuando ciertas condiciones de error particularmente extrañas nos impiden operarlo en forma normal. Para efectuar una inicialización, se debe entrar la orden
OPEN 15,8,15,"I":CLOSE15

VALIDATE (VALIDAR):

Este último comando del D.O.S sirve para "ordenar" un disco que, a raíz de numerosas operaciones de grabación y borrado, tiene una mala distribución de los bloques existentes en él. La única restricción a esta orden surje en aquellos discos que contienen archivos de tipo relativo, que son los mas usados por los sistemas administradores de bases de datos. En caso de tener un disco de este tipo NUNCA efectúe una VALIDATE sobre el mismo. Para ejecutar este comando ingrese la orden
OPEN 15,8,15,"V":CLOSE15

OTROS ITEMS DEL D.O.S.:

Como dijéramos al principio de esta nota, el canal de comando está preparado para entregarnos informes sobre sus condiciones de error. El manual de las disqueteras incluye une breve rutina BASIC para leer ese canal de error que nos informa el número codigo del error, su descripción en una breve leyenda; y la pista y sector en donde se detectó el mismo. Si se está empleando o desarrollando un programa de archivos es imprescindible contar con una subrutina que verifique este canal después de cada operación de entrada/salida. Al mismo tiempo, podemos utilizarla en modo directo para leer ese canal cada vez que vemos la luz roja de la disquettera titilar.

Por otra parte, existen en todo disco de demostración del 1541 dos

archivos de programa llamados "C-64 WEDGE" y "DOS 5.1" que sirven de gran ayuda. El primero, como su nombre lo indica, es una cuña que carga al segundo. Esta, una vez activado, brinda una serie de comandos directos que, en realidad, son una abreviatura de los aquí explicados (que pertenecen al D.O.S Versión 2.0); más algunos habitualmente no disponibles. En el listado 4, exhibimos la correlación entre los comandos del D.O.S 5.1 y el D.O.S 2.0.

Para sacar mayor provecho del WEDGE y el D.O.S. 5.1 recomendamos copiar ambos al principio de todo disquette nuevo y, cada vez que se enciende la máquina, cargarlos y correrlos para tenerlos siempre activos. Si, una vez cargado el D.O.S 5.1, se ejecuta una orden SYS 64738, no es necesario volver a cargar y correr el WEDGE, con sólo pulsar SYS 52224 se volverá a activar.

Adjunto a este artículo (listado 3) encontrarán Uds el programa "Inspector de Directorios" (no suena a cargo de funcionario?) que sirve para obtener mayor información sobre los archivos contenidos en un disco. Una vez tipeado y grabado en disco corra el programa colocando un disco en el drive. Una vez hecho esto pulse RETURN y el programa le hará conocer el nombre y tipo de cada archivo junto con la longitud del archivo en bloques (columna LAR); la primera pista (columna PI) y sector (columna SE) que éste ocupa y, si el archivo es relativo, la longitud de cada registro del mismo (columna LRG).

... Y ESTE ES EL BUS QUE SE VA:

Todos los comandos del D.O.S que ejemplificamos en este artículo están en la así llamada "forma abreviada", es decir, se invocan usando sólo la inicial del comando. Si bien no tiene mayor aplicación, esos comandos pueden ser invocados utilizando su nombre completo. Por ejemplo, para validar un disco se pueda ejecutar la sentencia OPEN 15,8,15,"V" cuyo efecto será idéntico a OPEN 15,8,15,"VALIDATE". Por otra parte, los comandos del D.O.S también se pueden ordenar por medio de sentencias PRINT# desde dentro de un programa y no en modo inmediato como los hemos ejemplificado. Para ello se deberá tener la precaución de abrir el canal de comando antes de efectuar cualquier comando por medio de un PRINT#. De lo contrario, el programa se detendrá con un mensaje de "FILE NOT OPEN ERROR" (el archivo no fue abierto).

De igual modo no se deben usar las sentencias tal cual las ejemplificamos en esta nota una vez que el archivo fue abierto. Si se intenta abrir un archivo que ya ha sido abierto el programa se interrumpirá con un "FILE OPEN ERROR" (el archivo ya está abierto).

CARLOS A. AY
y DANIEL H. MANOUCA

```
0 REM ********************************
1 REM OPCION 1: GRABA CADA VEZ QUE SE
2 REM           EJECUTA EL PROGRAMA.-
3 REM ********************************
4 :
5 NA$="PROGRAMA"
6 OPEN 15,8,15,"@:"+NA$:CLOSE15
7 SAVE NA$,8:VERIFY NA$,8
8 :
9 REM AQUI CONTINUA EL PROGRAMA
```

```
0 REM ********************************
1 REM OPCION 2: GRABA CADA VEZ QUE SE
2 REM           SE ORDENA GOSUB 100
3 REM           EN MODO DIRECTO O DESDE
4 REM           UN PUNTO DEL PROGRAMA.-
5 REM ********************************
70 :
80 :
96 :
100 NA$="PROGRAMA"
110 OPEN 15,8,15,"S:"+NA$:CLOSE15
120 SAVE NA$,8:VERIFY NA$,8
130 RETURN
131 :
144 REM AQUI SIGUE EL PROGRAMA
```

```
0 REM ********************************
1 REM OPCION 3: GRABA CADA VEZ QUE SE
2 REM           ORDENA GOTO 10000 EN
3 REM           MODO DIRECTO -
4 REM ********************************
9997 :
9998 REM AQUI TERMINA EL PROGRAMA
9991 :
9999 STOP
10000 NA$="PROGRAMA"
10010 OPEN 15,8,15,"S:"+NA$: CLOSE15
10020 SAVE NA$,8:VERIFY NA$,8
```

LISTADO COMPARATIVO DE COMANDOS D.O.S. 5.1 VERSUS D.O.S. 2.0

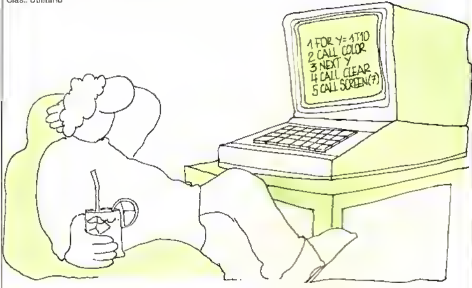
COMANDO DOS 2.0	COMANDO DOS 5.1	OBSERVACIONES
LOAD"PROGRAMA",8	/PROGRAMA	
LOAD"PROGRAMA",8,1	%PROGRAMA	LENG. DE MAQUINA
LOAD"PROGRAMA",8: RUN	↑PROGRAMA	AUTO-RUN
SAVE"PROGRAMA",8	←PROGRAMA	
SAVE"@: PROGRAMA",8	←@: PROGRAMA	SAVE & REPLACE
(NO EXISTE)	@ O >	LEE CANAL ERROR
(NO EXISTE)	@$ O >$	LEE DIRECTORIO
PRINT#15, "C: COPIA=ORIGINAL"	@C: COPIA=ORIGINAL	COPIA
PRINT#15, "I"	@I	INICIALIZA
PRINT#15, "N: NOMBRE, ID"	@NOMBRE, ID	FORMATEA
PRINT#15, "Q"	@Q	ANULA DOS 5.1
PRINT#15, "R: NUEVO=VIEJO"	@R: NUEVO=VIEJO	RENOMBRA
PRINT#15, "S: ARCHIVO"	@S: ARCHIVO	BORRA ARCHIVO
PRINT#15, "UI"	@U1	REESTABLECE DOS
PRINT#15, "V"	@V	VALIDA

ESTE LISTADO ASUME QUE ANTES DE LAS SENTENCIAS QUE EMPIEZAN CON PRINT#15 SE ORDENA UN OPEN 15,8,15 Y QUE LUEGO DE ELLAS SE EFECTUA EL CORRESPONDIENTE CLOSE 15.

```
0 rem ************************************
1 rem *                                  *
2 rem *    inspector de directorios      *
3 rem *                                  *
4 rem *                                  *
5 rem * copyright 1985, carlos a. ay     *
6 rem *                                  *
7 rem ************************************
8 poke53280,0:poke53281,0:poke648,3
9 print chr$(142):poke657,128:poke650,0
10 poke54296,15:poke36873,08
11 poke36872,172:poke36277,128+1
12 poke36278,16+1:a1=0:t=1:p=1
14 goto1000
20 poke781,f:poke782,c
30 poke783,peek(783)and254:sys65520
40 return
100 get#3,a$:a=asc(a$+chr$(0)):return
110 fori=0to13:poke781,i
120 sys39983:next:return
130 f=12:c=4:gosub20
140 print"{RVON}{YEL}"
120 print"FIN DEL DIRECTORIO...."
140 return
1000 print"{CLR}"
1010 fori=1to6
1020 c=0:gosub20:printchr$(160)
1030 c=10:gosub20:printchr$(170)
1040 next
1050 f=0:c=0:gosub20
1060 printchr$(111);
1070 fori=1to4:printchr$(183);:next
1080 printchr$(112)
1090 f=7:c=0:gosub20
1100 printchr$(180);
1110 fori=1to7:printchr$(175);:next
1120 printchr$(186)
1130 f=1:c=1:gosub20
1140 printchr$(75)chr$(54)chr$(52)
1150 f=2:c=4:gosub20
1160 printchr$(117)chr$(100)chr$(105)
1170 f=3:c=4:gosub20
1180 printchr$(103)chr$(32)chr$(104)
1190 f=4:c=4:gosub20:printchr$(106)
1200 printchr$(102)chr$(107)chr$(104)
1210 c=3:fori=5to6
1220 gosub20:printchr$(113)
12.  next
1240 f=0:c=12:gosub20
1250 fori=12to39:print"*";:next
1260 f=7:gosub20
1270 fori=12to39:print"*";:next
1280 fori=1to6
1290 c=12:gosub20:print"*"
1300 c=39:gosub20:print"*"
1310 next
1320 f=1:c=14:gosub20
1330 print"inspector de directorios"
1340 f=3:c=20:gosub20
1350 print"escrito por"
1360 f=5:c=10:gosub20
1370 print"carlos arteaga ay"
1380 f=9:c=2:gosub20
1390 print"coloque en la unidad numero";
1400 print" 8 el disco"
1410 f=11:gosub20
1420 print"cuyo directorio desea";
1430 print" inspeccionar."
1440 f=13:gosub20
1450 print"cierre la puerta de la";
1460 print" unidad y pulse"
1470 f=13:gosub20
1480 print"<return> para continuar."
1490 if a1=255then1490
1500 a1=a1+6
1510 fori=55296to55376step40
1520 forj=itoi+10
1530 pokej,a1
1540 next:next
1550 poke36274,33:poke34274,32
1560 geta$:ifa$=""then1490
1570 print"{CLR}"chr$(14)
1580 fori=0to4:readt$(i):next
1590 data "delo","sec.","prog."
1600 data "usuar.","relat."
1610 f=1:c=1:gosub20
1620 print"Nombre del disco:"
1630 f=2:gosub20
1640 print"Identificador:"
1650 f=3:gosub20
1660 print"D.O.S. Version:"
1670 f=0:gosub20
1680 print"Directorio"
1690 c=30:gosub20
1700 print"Bloques:"
1710 f=6:c=0:gosub20
1720 print"Nombre"
1730 c=18:gosub20
1740 print"Tipo"
1750 c=27:gosub20
1760 print"LAR OL DB LRG"
1770 f=7:c=0:gosub20
1780 fori=0to39:printchr$(163);:next
1790 f=10:gosub20
1800 fori=0to39:printchr$(163);:next
1810 f=18:c=1:gosub20
1820 print"Presione <RETURN> para ";
1830 print"ver la opciones"
1840 f=19:c=3:gosub20
1850 print"opciones "
1860 f=20:c=1:gosub20
1870 print"Presione <ESPACIO> para ";
1880 print"revisar el di-"
1890 f=21:c=3:gosub20
1900 print"rectorio de otro disco. "
1910 p=10:s=0:t=1
1920 open13,8,13
1930 open5,8,5,"#"
1940 print#13,"u1:"5;0;t;s
1950 gosub100:t=a:gosub100:s=a
1960 fori=2to103:gosub100:next
1970 n$="":fori=0to17
1980 gosub100
1990 ifa<>160thenn$=n$+a$
2000 next
2010 f=1:c=19:gosub20:print"{RVON}"n$
2020 id$="":fori=1to3:gosub100
2030 id$=id$+a$:next
2040 f=2:c=18:gosub20:printid$
2050 do$="":fori=1to2:gosub100
2060 do$=do$+a$:next
2070 f=3:c=17:gosub20:printdo$
2080 ift=0thengosub130:goto2340
2090 print#13,"u1:"5;0;t;s
2095 f=4:c=37:gosub20:printp;
2100 gosub100:t=a:gosub100:s=a:f=7
2105 p=p+1
2110 fork=0to7
2120 gosub100:tp=a and7
2130 gosub100:pl=a
2140 gosub100:se=a
2150 no$="":fori=0to15:gosub100
2160 ifa<>160thenno$=no$+a$
2170 next
2180 fori=0to2:gosub100:next:lo=a
2190 fori=22to27:gosub100:next
2200 gosub100:l1=a:gosub100:l=a
2210 l=256*lh+l1
2220 gosub100:gosub100
2230 ifno$=""then2310
2240 f=f+1:c=0:gosub20:printno$
2250 c=18:gosub20:printt$(tp)
2260 c=24:gosub20:printl
2270 c=30:gosub20:printpl
2280 c=33:gosub20:printse
2290 if tp<4then2310
2300 c=36:gosub20:printlo
2310 next
2320 getk$
2330 ifk$=chr$(13)thengosub110:goto2080
2340 ifk$<>chr$(32)then2320
2343 close5:close13
2350 fori=18to21:poke781,i
2360 sys39983:next
2370 print"{RVON}"
2380 f=19:c=1:gosub20
2390 print"Desea examinar otro disco";
2400 input"{C/RT}{C/RT}s{C/LF}{C/LF}{C/LF}{C/LF}";
2410 ifleft$(sn$,1)="s"thenrun
2420 f=21:c=1:gosub20
2430 print"Confirma salida";
2440 input"{C/RT}{C/RT}n{C/LF}{C/LF}{C/LF}{C/LF}";sn$
2450 ifleft$(sn$,1)="n"then2400
2460 end:rem sys64738
```

NUMERADOR AUTOMATICO DE LINEAS:

Comp.: Commodore 64
Cont.: 64K con disquetera o grabador a cassettes
Clas.: Utilitario



Este utilitario, escrito totalmente en lenguaje de máquina, numera automáticamente la próxima línea de un programa BASIC cuando se pulsa la tecla RETURN. Por defecto, el programa genera números de línea a partir del 0 con incrementos de 10, aunque estos valores se pueden modificar. Además, el numerador incluye una función especial que junto con el número de línea entrega la leyenda DATA, lo que resulta particularmente útil para programas con abundantes sentencias de este tipo.

Para cargar este utilitario, hay que ingresar y grabar en disco o cassette el cargador BASIC adjunto. Debemos correr este booter, que se encargará de colocar el utilitario a partir de la posición 49152 de memoria, y luego pulsamos NEW. A continuación, ingresamos en modo directo la sentencia SYS 49152, con lo que se activará el numerador. El siguiente cuadro detalla el efecto de las teclas de función afectadas por el numerador automático:

- f1 enciende y apaga la función de numeración automática.
- f3 enciende y apaga la función de leyenda DATA automáticamente
- f5 permite modificar el incremento a utilizar en la numeración automática
- f7 permite modificar la línea desde donde comienza la numeración automática.

Por ejemplo, cargamos y ejecutamos el booter y tipeamos NEW y SYS 49152. Después pulsamos f1 y RETURN, de inmediato aparecerá la línea número 10. Pulsamos una sentencia cualquiera, pulsamos RETURN y veremos la línea 10. Pulsamos f3 y la línea siguiente comenzará como 20 DATA. Pulsamos f3 nuevamente y la línea 30 será normal (es decir, sin leyenda DATA).

Si deseamos comenzar en la línea 1000 (o cualquier otra), pulsamos f7, ingresamos el número de línea deseado y tecleamos RETURN. Pulsamos RETURN otra vez y dicha línea será la próxima a ingresar. Si queremos modificar el incremento entre número de línea, pulsamos f5 e ingresamos el mismo. A partir de ese momento, las líneas estarán numeradas de acuerdo al incremento pedido. Para desactivar la función de numeración automática, pulsamos f1 nuevamente y el editor volverá a su funcionamiento normal. Para desactivar totalmente el numerador pulsamos RUN/STOP y RESTORE al mismo tiempo. Si no apagamos la C-64, SYS 49152 volverá a activar el numerador.



```
0 REM ***********************************
1 REM *                                 *
2 REM *      NUMERADOR AUTOMATICO       *
3 REM *                                 *
4 REM *  REVISTA K64, AGOSTO DE 1985     *
5 REM *                                 *
6 REM ***********************************
10 FOR1=49152TO49537:READA:POKE1,A:NEXT
49152 DATA 173,36,3,201,188,240,19,141,
189,192
49162 DATA 173,37,3,141,190,192,169,188
,141,36
49172 DATA 3,169,192,141,37,3,173,20,3,
201
49182 DATA 55,240,21,141,102,192,173,21
,3,141
49192 DATA 103,192,120,169,55,141,20,3,
169,192
49202 DATA 141,21,3,88,96,165,197,201,6
4,240
49212 DATA 40,205,127,193,240,35,141,12
7,193,201
49222 DATA 4,208,8,173,124,193,73,255,1
41,124
49232 DATA 193,201,5,208,8,173,125,193,
73,255
49242 DATA 141,125,193,201,6,240,31,201
,3,240
49252 DATA 3,76,49,234,169,70,32,153,19
2,141
49262 DATA 122,193,141,128,193,165,21,1
41,123,193
49272 DATA 141,129,193,162,255,108,0,3,
169,91
49282 DATA 32,153,192,141.126,193,173,1
28.193.141
49292 DATA 122,193,173,129,193,141,129,
193,162,255
49302 DATA 108,0,3,160,193,32,30,171,32
,96
49312 DATA 165,134,122,[illegible],32,115,0
,170,240
49322 DATA 243,162,255,134,58,144,1,96,
32,107
49332 DATA 169,169,0,133,198,165,20,96,
32,87
49342 DATA 241,8,133,251,201,13,240,4,[illegible]
65,251
49352 DATA 40,96,173,124,193,240,247,16
9,13,32
49362 DATA 210,255,165,214,141,130,193,
206,130,193
49372 DATA 134,252,132,253,173,123,193,
174,122,193
49382 DATA 133,98,134.99,142,128,193,14
1,129,193
49392 DATA 162,144,56,32,73,188,32,223,
189,162
49402 DATA 0,189,0,1,240,7,157,119,2,23
2
49412 DATA 76,251,192,169,32,157,119,2,
232,173
49422 DATA 125,193,208,29,134,198,173,1
22,193,24
49432 DATA 109,126,193,141.122,193,144,
3,238,123
49442 DATA 193,166,252.164,252,173,130,
193,133,214
49452 DATA 76,198,192,160,0.185,65,193,
240,8
49462 DATA 200,157,119,2,232,76,49,193,
76.18
49472 DATA 193,68,65,84,65,0.13,67,79,[illegible]
7
49482 DATA 73,69,78.90,79,32,69,78,32,[illegible]
6
49492 DATA 73,78,69,65.63,32,0,13,81,85
49502 DATA 69,32,86,65,76.79,82.32,73,7
8
49512 DATA 67,82,69,77,69,78,84.79,40,4
8
49522 DATA [illegible]
49532 DATA 0,0,10,64,0,0
```




CONTADOR DE BLOQUES LIBRES:

Comp.: Commodore 64
Conf. 64K con disquetera Commodore 1541
Clas. Utilitario



Este rápido utilitario en lenguaje de máquina ha sido preparado para residir en memoria junto con un programa BASIC y tiene como objetivo informar al usuario la cantidad de bloques existentes en el disquette alojado en el drive con número de dispositivo 8. Este contador será particularmente útil para decidir si existe espacio suficiente para grabar (SAVE) el programa BASIC que se encuentra en memoria, evitando el trabajo de cambiar disquettes o cargar el directorio que, si no se cuenta con el DOS 5.1 (ver disquette DEMO del 1541), borrará el programa BASIC. Una vez cargado en memoria, el utilitario puede ser llamado tecleando en modo directo. SYS 49152. ATENCION: antes de cargar este utilitario verifiquemos que el área RAM que comienza en 49152 no se halla ocupada por algún otro programa. De residir en ella un utilitario tal como el numerador automático el C-64 sufrirá un "crash" irrecuperable (habrá que reencenderlo para que vuelva a comportarse correctamente).

```
100 REM *********************************
105 REM *                               *
110 REM *  CONTADOR DE BLOQUES LIBRES   *
115 REM *                               *
120 REM * REVISTA K64, AGOSTO DE 1985.  *
125 REM *                               *
130 REM *********************************
140 PRINT"{CLR}":POKE53281,0
145 PRINT"ESTE PROGRAMA CARGA UNA RUTINA
EN "
150 PRINT"LENGUAJE DE MAQUINA EN MEMORIA.
"
160 FORX=49152TO49307:READA:B=B+A
165 NEXT:IFB<>20757THEN220
170 RESTORE
175 FORX=49152TO49307:READA
180 POKEX,A:NEXT:PRINT
185 PRINT"LA RUTINA SE EJECUTARA"
190 PRINT"TECLEANDO 'SYS 49152'"
195 PRINT"Y PRESIONANDO <RETURN>, A CONTI
NUACION"
200 PRINT"CONTARA LA CANTIDAD DE BLOQUES
LIBRES"
205 PRINT"DISPONIBLES EN EL DISCO DEL DRI
VE 8."
210 PRINT"{C/DN}{C/DN}VERIFIQUE QUE LA DI
SQUETERA ESTE"
215 PRINT"ENCENDIDA''":END
220 PRINTCHR$(28)
225 PRINT"{CLR}EXISTE UN ERROR EN UNA"
230 PRINT"SENTENCIA 'DATA'."
235 PRINT"{C/DN}VERIFIQUE EL LISTADO Y"
240 PRINT"CORRIJA VALORES INCORRECTOS."
245 END
250 DATA 169,2,162,8,160,2,32,186,255
255 DATA 169,1,162,137,160,192,32,189
260 DATA 255,32,192,255,32,68,229,162
265 DATA 2,32,198,255,169.0,141,136
270 DATA 192,169,2,141,134,192,32,228
275 DATA 255,32,228,255,32,228,255,141
280 DATA 135,192,32,228,255,32,228,255
285 DATA 32,228,255,32,228,255,174,134
290 DATA 192,224,18,240,18,24,109,135
295 DATA 192,141,135,192,173,136,192
300 DATA 105,0,141,136,192,174,134,192
305 DATA 232,142,134,192,224,36,208,211
310 DATA 169,141,32,210,255,32,210,255
315 DATA 174,135,192,173,136,192,32,205
320 DATA 189,169,32,32,210,255,169,138
325 DATA 160,192,32,30,171,32,204,255
330 DATA 169,2,32,195,255,96,0,0
335 DATA 0,36,66,76,79,81,85,69,83,32
340 DATA 83,73,78,32,85,83,65,82,46,0
```

AHORRO DE MEMORIA DE SU 1000/1500

(Parte III)

Continuando con el número anterior, ofrecemos la tercera y última parte de este interesante tema.

¿CUANDO CONSUMIMOS 24 BYTES?:

10 IF A=1 THEN GOTO 9 pero, 10 IF A=1 THEN GOTO 10 consume 25 bytes
El uso de "< =";"<";">" y " < > " en líneas como esta, también dan un consumo idéntico.
10 IF A=1 OR B < 2 THEN GOTO 9 consume 34 bytes, de modo que "OR B < 2" insume 10 bytes
10 IF NOT A=1 THEN GOTO 9 consume 25 bytes por lo que NOT solamente consume 1 byte
10 IF A=1 THEN GOSUB 9 también consume 24 bytes

IMPRIMIENDO ORDENADAMENTE

Existen dos funciones asociadas a le sentencia PRINT que imprimen en un lugar determinado de la pantalla, estas funciones son AT y TAB, veamos cada una de ellas

PRINT AT:

10 PRINT AT 1,1,"A" consume 26 bytes de modo que AT 1,1," insume 17 bytes, en gran parte por utilizar los números, en cambio si escribimos 10 PRINT AT L,L,"A" se consumirán solamente 14 bytes

TAB:

10 PRINT TAB 1;"A" consume 18 bytes, lo cual hace que "TAB 1;" tome 9 bytes
10 PRINT TAB X,"A" en este caso sólo consume 12 bytes.

COMPARACION ENTRE PRINT AT Y SENTENCIAS PRINT VACIAS

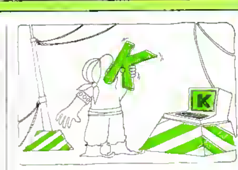
Si tenemos:
10 PRINT AT 4,1;"A" consume como dijimos 26 bytes, pero

```
10 PRINT
15 PRINT
20 PRINT          consume 27 bytes
25 PRINT "A"
```

Como se puede apreciar, es muy importante utilizar sentencias PRINT vacías para dejar una o dos líneas en blanco, pero para tres o más, es económico la utilización del PRINT AT.
También podemos tener el caso 10 PRINT AT 2,5;"A" con 26 bytes de consumo, reemplazar por.

```
10 PRINT
15 PRINT TAB 5,"A" donde el consumo es de 24 bytes
```

Si se emplean nombres de variables en vez de números, entonces el imprimir ordenadamente puede ser más conveniente y efectivo, así por ejemplo:
10 PRINT AT X Y;"A" solamente consumirá 14 bytes



SUBCADENAS

10 PRINT A$ tiene un consumo de 8 bytes, en cambio
10 PRINT AS (TO 9) consume 18 bytes de modo que "(TO 9)" insume 10 bytes
10 PRINT AS (TO 10) nos consume 19 bytes debido al caracter extra
10 PRINT A$ (1 TO 9) consume 25 bytes
10 PRINT AS (1 TO) también consume 18 bytes
Aquí otra vez, podemos ahorrar memoria utilizando nombre de variables en lugar de números
10 PRINT A$ (A TO B) consume solamente 13 bytes, comparándolo con los 25 bytes

RESUMEN

Una vez analizados estos consumos es importante tener a mano una pequeña subrutina de modo tal, que cuando se está programando nos indique la longitud del programa, es decir, la cantidad de bytes utilizados
La subrutina que se debe agregar es
9999 PRINT "LONGITUD DEL PROGRAMA –"; PEEK 16396+256*PEEK 16397-16596; "–BYTES"
donde – indica un espacio
Si esta subrutine se ejecuta, el resultado es "0 BYTES". La manera de trabajar con ella es la siguiente.
Una vez que se ingresó un programa o en un estado intermedio, antes de correrlo hacemos un GOTO 9999, y en pantalla aparecerá la cantidad de bytes insumidos
Les deseamos que tengan muchos éxitos.

Ing. JULIO JOSE PUTRUELE
Ing. MIGUEL ANGEL MAUBRO

CONCURSO TRIMESTRAL





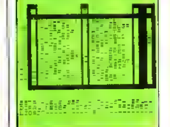
PRODE

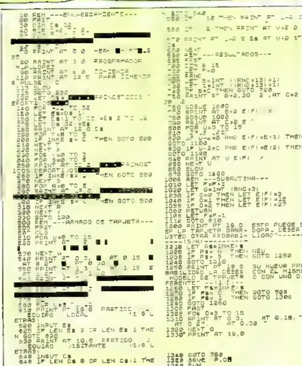
Comp. CZ1000/1500 TK83/85
Conf. 16 K
Clas. Entretenimiento
Autor: Manuel A. López

LISTA DE VARIABLES

Z LOOP: Tiñe la pantalla de negro.
Y LOOP: Controla los cambios del título.
X LOOP: Controla el movimiento del título.
W LOOP: Dibuja las columnas de la tarjeta.
V LOOP: Permite el ingreso de los 13 partidos.
U LOOP: Selecciona los 13 resultados.
T LOOP: Permite el comienzo del juego.
S LOOP: Imprime durante un momento el título en video normal.
R LOOP: Imprime durante un momento el título en video inverso.
O LOOP: Borrado de cruces.

PANTALLA





```
  10 REM ---ENCABEZAMIENTO---
  20 FAST
  30 DIM C$ 32
  [illegible]
  70 PRINT AT 0,0 [illegible]
  80 PRINT AT 3,0 "PROGRAMADOR M.A.LOPEZ"
  90 PRINT AT [illegible] COMIENZA
 110 PRINT AT 12,5 "PARA COMENZAR PULSE C"
 113 SLOW
 115 GOTO 230
 120 LET [illegible] PRINCIPIOS [illegible] DEPORTIVOS [illegible]
 140 FOR [illegible] TO 32
 150 LET E$=D$ [illegible]
 160 LET D$(1 TO 31)=E$(2 TO 32)
 170 LET D$(32)=E$
 180 PRINT AT 12,0;D$
 190 FOR T=0 TO 5
 200 IF INKEY$="C" THEN GOTO 500
 210 NEXT T
 220 NEXT X
 230 FOR Y=0 TO 3
 240 FOR S=0 TO 5
 250 PRINT AT [illegible] PRINCIPIOS DEPORTIVOS [illegible]
 260 IF INKEY$="C" THEN GOTO 500
 270 NEXT S
 280 FOR R=0 TO 5
 290 PRINT AT [illegible]
 300 IF INKEY$="C" THEN GOTO 500
 310 NEXT R
 320 NEXT Y
 330 GOTO 120
 500 REM ---ARMADO DE TARJETA---
 505 FAST
 507 CLS
 510 FOR W=0 TO 15
 520 PRINT AT [illegible]
 530 NEXT W
 540 PRINT AT 0,3; AT 0,15; AT 0,27; AT 0,30; [illegible]
 550 PRINT AT 1,3;"L"; AT 1,15;"E"; AT 1,27;"V"; AT 1,30 [illegible]
 [illegible]
 600 FOR V=[illegible]
 610 PRINT AT 19,0;"PARTIDO [illegible] EQUIPO LOCAL (1 A 9 LETRAS)"
 620 INPUT E$
 625 IF LEN E$>9 OR LEN E$<1 THEN GOTO 620
 630 PRINT AT 19,0;"PARTIDO [illegible] EQUIPO VISITANTE (1 A 9 LETRAS)"
 640 INPUT C$
 645 IF LEN C$>9 OR LEN C$<1 THEN [illegible]
 [illegible]
 660 IF [illegible] THEN PRINT AT V+2 [illegible]
 670 PRINT AT [illegible]; AT V+2,17;C$
 680 NEXT V
 [illegible] REM ---RESULTADOS---
 [illegible] FAST
 [illegible]
 [illegible] LET F=0
 [illegible]
 [illegible] LET B=INT ((RND*13)+1)
 [illegible] LET C=INT ((RND*13)+1)
 [illegible] IF B=C THEN GOTO 720
 [illegible] PRINT AT B+2,30;"2"; AT C+2,32;"2"
 760 GOSUB 1000
 770 PRINT AT B+2,E(F);"X"
 780 GOSUB 1000
 790 PRINT AT C+2,E [illegible]
 800 FOR U=3 TO 15
 810 [illegible]
 820 IF U-2=B AND E(F)=E(1) THEN GOTO 1100
 830 IF U-2=C AND E(F)=E(2) THEN GOTO 1100
 840 PRINT AT U,E(F);"X"
 850 NEXT U
 860 SLOW
 870 GOTO 1200
1000 REM ---SUBRUTINA---
1010 LET F=F+1
1020 LET O=INT (RND*3)
1030 IF O=0 THEN LET E(F)=3
1040 IF O=1 THEN LET E(F)=15
1050 IF O=2 THEN LET E(F)=27
1060 RETURN
1100 LET F=F-1
1110 GOTO 810
1200 PRINT AT 19,0;"ESTA PUEDE SER LA TARJETA GANADORA. DESEA USTED OTRA PROBABILIDAD?-------------(S/N)-----------"
1210 LET P$=INKEY$
1220 IF P$="N" THEN NEW
1230 IF P$="S" THEN GOTO 1250
1240 GOTO 1210
1250 PRINT AT 19,0;"SU NUEVA PROBABILIDAD LA DESEA CON EL MISMO MODELO DE TARJETA O CON UNO DIFERENTE? --(1 2)--"
1260 LET F$=INKEY$
1270 IF F$="0" THEN GOTO 500
1280 IF F$="1" THEN GOTO 1300
1290 GOTO 1260
1300 FAST
1305 FOR O=3 TO 15
1310 PRINT AT O,3;" "; AT O,10;" "; AT O,27;" "; AT O,30;" "
1320 NEXT O
1330 PRINT AT 19,0
1340 GOTO 700
1350 SAVE "P.O"
1360 RUN
```

AS ALFANUMERICA: Ingresa la contestación a una nueva probabilidad
BS ALFANUMERICAS. Ingresa el nombre del equipo local
CS ALFANUMERICAS. Ingresa el nombre del equipo visitante
DS ALFANUMERICAS (Dimensionada). Guarda el título
ES ALFANUMERICAS Guarda el Primer caracter de DS
FS ALFANUMERICAS Ingresa la contestación al nuevo tipo de tarjeta
F. Controla el número de resultado.
E (Dimensionada): Guarda los resultados
TIPO DE PROGRAMA Juego de Azar
CAPACIDAD DE MEMORIA. 2,9 K
MAQUINAS EN QUE PUEDE SER CORRIDO: TK 85 y similares cuya configuración mínima de memoria sea de 16 K
INSTRUCCIONES. Sólo hay que obedecer el programa
TIPO DE CINTA. El programa fue grabado con una TK 85 en una cinta del tipo NORMAL sin ningún tipo de ecualización especial
NUMERO DE CUENTAVUELTAS 063.

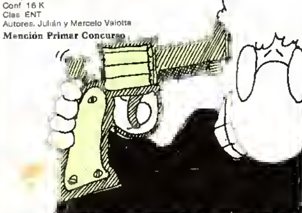
COMENTARIO DE LINEAS:

10-113 Arman la presentación del juego
115-330 Generan el movimiento y el cambio de video del título
500-575 Arman la tarjeta
580-680. Permiten la entrada de los nombres de los equipos.
690-790. Seleccionan los dobles al azar y los imprimen
800-860. Eligen los resultados posibles los compara con los resultados de los dobles ya elegidos, si el resultado es el mismo al igual que el resultado elige un nuevo resultado, de lo contrario imprime
700: Envía a ejecutar las preguntas
1000-1110: Subrutina de selección de resultados.
1200-1290: Pregunta por nuevas probabilidades y tipo de tarjeta.
1300-1320: Borrado de cruces.
1330: Borrado de pregunta.
1340: Manda una nueva selección de resultados
1350: Graba el programa.
1360: Corre el programa

RULETA RUSA

Comp TK 83/85, CZ 1000/1500
Conf 16 K
Clas ENT
Autores. Julián y Marcelo Valotta

Mención Primer Concurso



El Programa en el cual esta presente el azar, consta de gráficos en una cadena de la forma A$ (por ej.) De esta manera, los revólveres y la tumba que aparecen durante el juego, están incluidos dentro de las correspondientes cadenas
En el listado del programa las variables AS, BS, y CS no están definidas, por lo tanto una vez que se haya terminado de tipear el programa y antes de ejecutarlo, hay que darle el valor a cada una de estas variables de la siguiente manera: LET nombre de variable "contenido", sin número de línea.

Por este motivo nunca hay que hacer correr el programa con el comando RUN. En cambio hay que usar GOTO 1.

COMO FUNCIONA EL JUEGO:

Al empezar, pregunta el número de jugadores, el número de rondas y el nombre de cada uno de los participantes
Por turno los jugadores se deberán arriesgar -o no- a "apretar" el gatillo de la pistola que aparece en pantalla.
A la pregunta "se arriesga", uno puede responder presionando "S" (si) ó "n" (no) si contestamos afirmativamente, nos preguntará cuál es el número que elegimos (de 1 a 6), simulando así el tambor de la pistola de 6 tiros
Presionaremos a continuación el número que sea de nuestro agrado y es aquí donde se hace presente el azar
La computadora elige entonces -a su vez- un número del 1 al 6, y si este coincide con el elegido por nosotros...¡PUM! ...uno menos en el juego (aparecerá una tumba y la computadora anunciará nuestro final).
Si a la pregunta "se arriesga..." se contesta negativamente (pulsando N), la computadora nos tildará de cobardes.
A continuación se repiten los pasos anteriores con el siguiente participante, así hasta terminar la primer ronda.
Se trata de obtener la máxima cantidad de puntos, consiguiéndose estos puntos, al oprimir el disparador y no ser víctimas de la imaginaria bala alojada en el tambor de la pistola. un disparo fallido un punto.
Si por casualidad sucumbimos en el intento, la computadora nos pasará por alto en las siguientes ron-



CONCURSO TRIMESTRAL

das aunque puede darse el caso de que ningún otro participante, al terminar el juego, pueda alcanzarlo en la tabla de posiciones hasta el final, quedando así como el vencedor.
Alguien puede llegar a tener la posibilidad de probar suerte dos o más veces en una misma ronda, aumentando así el puntaje.
Al final de cada ronda se muestra la tabla de posiciones, donde aparecen todos los participantes ordenados de acuerdo a su cantidad de puntos.

Y se repiten todos los pasos anteriores, hasta concluir el número de rondas elegido al principio.

PANTALLA



```
   1 REM ** MARCELO   JULIAN **
   2 REM ****   .ALOTTA    ****
  10 REM  RULETA RUSA
  20 REM  BY ME! 1985
  30 PRINT AT 0 0.E$
  53 PRINT AT 17,0  CUANTOS JUEG
AN "
  54 INPUT A
  55 PRINT AT 17 0  P CUANTAS RO
NDAS DESEA JUGAR "
  58 INPUT B
  70 DIM O$(A 10)
  72 LET O20=0
  74 LET O40=0
  80 DIM P(A)
  82 DIM M(A)
 100 PRINT AT 17,0  INGRESE C U
SU NOMBRE...,
 120 FOR G=1 TO A
 140 INPUT O$(G)
 142 LET P(G)=0
 144 LET M(G)=0
 200 NEXT G
 400 CLS
 500 FOR X=1 TO B
 505 PRINT AT 10 10 "  ", RONDA
 506 PAUSE 20
 550 FOR Y=1 TO A
 555 IF M(Y)=1 THEN GOTO 5100
 600 PRINT AT 0 0 P$
 605 IF O20=1 THEN GOTO 1000
 610 PRINT AT 17,0  SE ARRIESGA
   O$(Y)   ?
 630 IF INKEY$= S  THEN GOTO 100
0
 649 IF INKE $= N  THEN GOTO 500
 650 GOTO 630


 970 PRINT AT 20 0    COMPRIE
 980 PAUSE 50
 990 CLS
 992 GOTO 5100
1000 PRINT AT 10,0  ELIJA UNA TE
CLA DE 1 A 6
1020 LET R$=INKEY$
1040 IF CODE R$ 28 AND CODE R$
E THEN GOTO 2000
1050 GOTO 1020
2000 LET O=INT (RND*6+1)
2020 IF CODE R$=O+28 THEN GOTO 5
000
2050 CLS
2070 PRINT AT 5 10  CLICK...
2080 PRINT AT 10 2  FELICIDADES
 GANO 1 PTO
2085 LET P(Y)=P(Y)+1
2090 PRINT TAB 3, YA COSECHO   P
(Y)   PTS
2100 LET O20=INT (RND*3+1)
2200 IF O20  1 THEN PAUSE 30
2220 IF O20  1 THEN GOTO 5100
2300 PRINT AT 13 0  SE QUIERE AR
RIESGAR DE NUEVO ?
2310 IF INKEY$= S  THEN GOTO 600
2311 IF INKEY$= N  THEN LET O20=
0
2320 IF INKEY$= N  THEN GOTO 500
0
2330 GOTO 2310
5000 CLS
5010 FOR N=10 TO 1 STEP -1
5020 PRINT AT N 8,
5022 PRINT AT N+1 9
5023 PRINT AT N+2 9
5024 PRINT AT N+3 9
5025 PRINT AT N+4 9
5026 PRINT AT N+5 9
5027 PRINT AT N+6 9,
5028 PRINT AT N+7 9
5030 PRINT AT N+8,9,
5035 NEXT N
5055 PRINT AT 2 2
5058 PRINT AT 17 6  LMAL  E
5070 PRINT AT 19 7  UNA MENOS  .
5084 PAUSE 30
5087 LET M(Y)=1
5090 CLS
5100 NEXT Y
5300 CLS
5400 PRINT AT 10 5  TERMINO LA R
ONDA   X
5410 PAUSE 20
5420 CLS
5500 FOR C=1 TO 10
5510 PRINT AT 2,7  TABLA DE POSI
CIONES
5520 FOR U=1 TO 2
5522 NEXT U
5530 PRINT AT 2,7  TABLA DE POSI
CIONES
5540 NEXT C
5542 PRINT
5600 LET S=0
5602 LET O40=O40+10
5603 FAST
5604 PRINT  POSIC.   NOMBRE  TA
B 16  PTS     CONDIC.
5606 PRINT  ======   ====== TA
B 10 ===,      =======
5608 PRINT
5620 FOR S=O40 TO 0 STEP -1
5700 FOR T=1 TO A
5720 IF M(T)=0 THEN LET P$="VIVO
5740 IF M(T)=1 THEN LET P$= MUER
TO
6000 IF P(T)<S THEN LET E=E+1
6010 IF E<1 THEN LET O40=P T
6020 IF P(T)<S THEN PRINT S TAB
7 O$(T),TAB 15 P(T) TAB 25 P$
6040 NEXT T
6050 NEXT S
6070 SLOW
7000 PAUSE 500
7500 CLS
8000 NEXT X
9000 PRINT
9010 PRINT
9012 PRINT  TERMINO EL SUPLICIO.
..
9014 PRINT  FELICIDADES AL VENCE
DOR
9016 PRINT TAB 10  .PRESS ANY KE
Y.
9020 PAUSE 9000
9050 GOTO 2
9100 SAVE  RULETA R.SA
9120 GOTO
```

SALVAR VIDAS

Comp CZ1000/1500 TK83/85
Conf 16 K
Clas Entretenimiento
Autor Julio Moreno



PANTALLA





```
   1 REM   ***  SALVAR VIDAS ****
   2 REM   *******************
   3 REM   *    ENVIADO  POR     *
   4 REM   ****  JULIO MORENO ***
   5 PRINT AT 10 3  BUSCA UNA LL
     ...TRAS (10   DISPONES DE
     ...ISARD 31    LOCERAS
     ...PASAR P     ...CA
     ...LIAR LA     PERSONA TE
     ...ON LAS      TECLAS 7 6

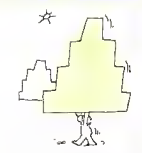
     PAUSE 500

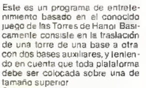
     ... L=1 7, 60
     PRINT AT _ 0
     NEXT L
     FOR U=1 TO 10
     PRINT AT 10 30
     PRINT AT 10 30
     NEXT A
     FOR H=0 TO 21
  53 PRINT AT H 2
  55 NEXT H
```



TORRES de HANOI

Comp: CZ1000/1500 TK83/85
Conf 16K
Clas Entretenimiento
Autor Miguel García





Este es un programa de entretenimiento basado en el conocido juego de las Torres de Hanoi. Básicamente consiste en la traslación de una torre de una base a otra con dos bases auxiliares, y teniendo en cuenta que toda plataforma debe ser colocada sobre una de tamaño superior.

Carece de instrucciones porque es sencillo y porque además el jugador puede ver cómo la máquina resuelve el problema planteado.

La codificación corresponde al lenguaje BASIC de la micro Sinclair ZX81.

La memoria requerida para su funcionamiento es de 16 K.

En la línea 20 se elige el nivel de la torre a trasladar con un tope de 9 plataformas. Desde la línea 30 hasta la 270 se cargan las variables y se presenta la torre. En la línea 280 se pide una opción:

a) "Vos (V)" para que el jugador pueda probar su capacidad. (600 a 770) Se piden los datos para el traslado, se validan dichos datos y si fueren adecuados se establece el cambio en la subrutina (900 a 970). El récord equivale a la cantidad de movimientos indispensables.

b) "Yo (Y)" para que la máquina resuelva el problema por sí misma (315 a 590). Para resolverlo, almacena en un vector (V) los cambios necesarios (320 a 510). Después ejecuta los cambios valiéndose de la subrutina mencionada (520 a 560). Está claro que la máquina lo hace con la menor cantidad de movimientos.

c) "Otra torre (T)", si se desea una torre distinta por parecerle demasiado fácil o demasiado difícil.

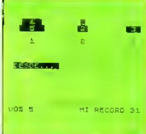
En todos los casos se vuelve a las opciones de la línea 280.

Para dar una idea de lo que significa resolver una torre de Hanoi, se definen los movimientos indispensables como $2^n - 1$ con n igual a la altura de la torre. Eso significa que resolver una torre de nivel 9 equivale a 511 movimientos como mínimo: esto supone para el jugador entrar 1022 datos.

Aunque el lenguaje BASIC no es recursivo puede, de vez en cuando, resolver problemas de recursividad.



PANTALLA





PANTALLA





CONCURSO TRIMESTRAL



EDUCAR

Comp CZ2000 TK 90 X TS 2068
Conf 16 K
Clas Educativo
Autor Carlos E Silva



El programa comienza pidiendo el nombre del usuario, y e allí en adelante se dirigirá a éste por su nombre. Si se desea modificar alguno de los temas, se debe ingresar en vez del nombre la Palabra CLAVE. Si se desea modificar todos los temas, se debe ingresar CLAVE TOTAL. Para archivar los nuevos datos se debe ingresar ARCHIVA. En todos los casos se deben seguir las instrucciones de la Pantalla. La Pregunta y la opción pueden tener hasta 13 caracteres.

El título de cada tema puede tener hasta 32 carecteres. Pero el menú queda poco estético si tiene más de 25

PANTALLA

```
 1 CAPITALES DE AFRICA
 2 CAPITALES DE AMERICA
 3 CAPITALES DE EUROPA
 4 CAPITALES DE ASIA
 5 CIUDADES DE AMERICA
 6 CIUDADES DEL MUNDO
 7 CAPITALES PROVINCIALES
 8 CIUDADES ARGENTINAS
 9 CHACO - CAP. DEPTOS. 1
 0 CHACO - CAP. DEPTOS. 11
```



```
  10 SLOW
  20 PRINT "[illegible]"
  30 INPUT N
  40 IF N<0 OR N>1 THEN GOTO 30
  50 LET L=2**N-1
  60 LET M=N+1
  70 DIM Y(3)
  80 DIM O$(3,6,9)
  90 CLS
  95 LET U=0
 100 LET Y(1)=1
 110 LET Y(2)=N
 120 LET Y(3)=N
 130 LET O$(1,1)=
 140 LET O$(1,2)=
 150 LET O$(1,3)=
 160 LET O$(1,4)=
 170 LET O$(1,5)=
 180 LET O$(1,6)=
 190 LET O$(1,7)=
 200 LET O$(1,8)=
 210 LET O$(1,9)=
 225 FOR K=1 TO N
 235 PRINT AT K,1 O$ 1
 245 NEXT K
 250 PRINT AT N+2 5 1
 255 PRINT AT N+2 15 2
 270 PRINT AT N+2 25 3
 280 PRINT AT 16 6 VOS(U) YO
(Y) OTRA TORRE (T)
 290 IF INKEY$="T" THEN RUN
 300 IF INKEY$="V" THEN GOTO 500
 310 IF INKEY$<>"Y" THEN GOTO 28
0
 315 FAST
 320 LET T=2*(2**N-1)
 330 DIM V(T)
 340 FOR I=1 TO N
 350 LET J=2**I-1
 360 LET V(J)=1
 370 IF I/2-INT (I/2)=0 THEN GOT
O 400
 385 LET V(J+1)=2
 390 GOTO 410
 400 LET V(J+1)=3
 410 LET X=0
 420 FOR J=2**I+1 TO 2**(I+1)-2
 430 LET X=X+1
 440 IF I/2-INT (I/2)=0 THEN GO
TO 470
 450 LET V(J)=V(X)+1
 460 GOTO 480
 470 LET V(J)=V(X)+2
 480 IF V(J)=5 THEN LET V(J)=2
 490 IF V(J)=4 THEN LET V(J)=1
 500 NEXT J
 510 NEXT I
 515 SLOW
 520 FOR I=1 TO T STEP 2
 530 LET A=V(I)
 540 LET B=V(I+1)
 550 GOSUB 900
 560 NEXT I
 570 PRINT AT 15 0 "LO ME CONSE
GUIDO EN " U " INTENTOS"
 580 PAUSE 200
 590 GOTO 70
 600 PRINT AT 15 0 "VOS U"
    AT 16 15 "MI RECORD"
 610 PRINT AT N+6 2 "[illegible]"
 620 INPUT A
 630 PRINT AT N+8 10 "[illegible]"
 640 INPUT B
 650 IF A<1 OR A>3 THEN GOTO 610
 660 IF B<1 OR B>3 THEN GOTO 610
 670 IF Y(A)<>M THEN GOTO 690
 680 GOTO 610
 690 IF Y(B)=M THEN GOTO 710
 700 IF CODE (O$(A,Y(A),5))>CODE
 (O$(B,Y(B),5)) THEN GOTO 610
 710 GOSUB 900
 720 PRINT AT 18 5 U
 730 IF Y(2)=1 OR Y(3)=1 THEN GO
TO 750
 740 GOTO 610
 750 PRINT AT 20 5 "COMPARE LO
S RESULTADOS"
 760 PAUSE 250
 770 GOTO 70
 900 REM "[illegible]"
 910 PRINT AT [illegible]
 920 LET Y(B)=Y(B)-1
 930 PRINT AT Y(B) B*10-9 O$(A
,Y(A))
 940 LET O$(B,Y(B))=O$(A,Y(A))
 950 LET Y(A)=Y(A)+1
 960 LET U=U+1
 970 RETURN
```

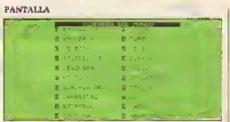
METEOROS

Comp CZ2000 TK90X TS2068
Conf 16 K
Clas Entretenimiento
Autor: Esteban Pellacani

```
   1 REM ++++++++++++++++++++++++++
   2 REM +                        +
   3 REM + HECHO EN QUILMES POR   +
   4 REM +                        +
   5 REM +  ESTEBAN PELLACANI     +
   6 REM +                        +
   7 REM ++++++++++++++++++++++++++
  10 CLS
  15 PAUSE 150
  20 GOSUB 200
  30 FOR I=1 TO 4: PRINT AT b(I),
c(I);: NEXT I
  40 LET v=v+(INKEY$="0" AND v<2
3)-(INKEY$="1" AND v>3)
  50 PRINT AT 20,v; INK 6;v$
  60 FOR x=1 TO 4
  70 IF b(x)<20 THEN GOTO 120
  80 LET b(x)=4: LET c(x)=[illegible]*3+1
NT (RND*11)
  90 LET p=p+1: PRINT AT 0,6;p
 100 IF p=25 THEN LET s=2
 105 IF p=50 THEN GOTO 500
 110 GOTO 130
 120 LET b(x)=b(x)+s
 130 SOUND .01,17: PRINT AT b(x),
c(x); INK (.5)
 140 IF b(x)<20 THEN GOTO 160
 150 IF (c(x)>v-2 AND c(x)<v+2)
THEN SOUND 2,20: GOTO 510
 160 NEXT x
 170 GOTO 30
 200 PAPER 5: CLS
 210 LET v$=CHR$ 134+CHR$ 14
0+CHR$ 140+CHR$ 137
 220 LET s$=CHR$ 140
 230 LET s=1: LET p=0: LET v=1
 240 DIM b(4): DIM c(4)
 250 FOR I=1 TO 4: READ b(I),
READ c(I): NEXT I
 260 PRINT AT 0,0;"PUNTOS: 0"
 270 RETURN
 280 DATA 16,4,12,8,4,16,8,20
 500 FOR g=1 TO 50: SOUND .01,5:
 NEXT g: FOR w=1 TO 4: FOR x=1 T
O 7: SOUND .01,x+15: BORDER x: P
APER x: CLS : NEXT x: NEXT w
 510 CLS : PRINT PAPER 4; IN
K AT 11,5;"FELICITACIONES": PAPE
R 1; AT 19,5;"Muy buen puntaje"
 PAPER 7: INK 0: p
 600 INPUT PAPER 4; INK 1;"Qui
ere jugar nuevamente (S/N)? ";q$
 610 IF q$="s" OR q$="S" THEN RU
N
 620 IF q$="n" THEN STOP
1000 STOP
9000 SAVE "METEOROS" LINE 10
```

PANTALLA

PANTALLA



CORREO • CONSULTAS

En esta sección atendemos todas aquellas consultas y sugerencias que nuestros lectores deseen realizar. Para ello sólo debe dirigirse a esta redacción, sección "Consultas".

SAVE

En primer lugar deseo felicitarlos por su revista y pedirles que sigan publicando programas para Sinclair/Tanto 1000 como 1500.

Una pregunta sobre la introducción en cassettes: ¿Se puede tipear primero lo que se desea introducir y una vez terminado, recién introducirlo en el cassette? Esto no lo tengo muy en claro ya que el manual no es muy explícito para principiantes.

¿Dónde puedo conseguir el Nº 1?

Horacio Mercado
ROSARIO SANTA FE

K-64

La computadora grabará en cassette usando la función SAVE, aquello que tenga en su memoria en ese momento, así lo haya tecleado o lo haya cargado antes de otro cassette. LAMENTABLEMENTE EL Nº 1 ESTA AGOTADO

CIRCUITO

He armado el sencillo circuito que ustedes publicaron en este número, pero no he obtenido los resultados necesarios, en una palabra no funciona. Hice un montón de pruebas dentro de mis conocimientos de electrónica, pero sigue negativo. A lo mejor hay algún valor de capacitor que no coincide. El circuito es el de DECK (omití mencionarlo)

K-64

El circuito al que se refiera, funciona bien y no hay omisiones o errores de impresión.

Sin embargo, aquí van unos consejos para que pruebe

* La tensión de alimentación debe ser de 5V, estabilizados

* La señal de entrada debe tener por lo menos 0,5 Vpp.

* Deben unirse las salidas de ambos canales del deck.

* Puede intentar aumentar la ganancia o sensibilidad del circuito, cambiando R3 A 1800 OHM

* Verifique que el resto de los componentes esté en buen estado, y conectados correctamente según su polaridad.

Esperamos que con estos datos pueda resolver el problema que comenta. Hasta pronto

ADOLFO L. de ARRIBA
Pergamino - Bs. As.

CARACTERES GRAFICOS

En el número 3, en el programa AM Spriter y Tragamonedas, hay una serie de líneas en las que hay caracteres gráficos especiales de Commodore que, supongo, por una cuestión de impresión, no resultan legibles y en el caso de Tragamonedas no se ve el efecto del programa. ¿Qué debo hacer?

Alberto Martinez
Florida

K 64

Ese es un problema de compatibilidad de los caracteres gráficos especiales de Commodore y la impresora que utilizan Carlos Ay y Daniel Manduca para listar los programas que presentan en la revista.

Con respecto al AM Spriter, los caracteres gráficos que no se ven son intrascendentes y no afectan al desarrollo y ejecución del programa.

En relación con Tragamonedas, publicamos en esta página una modificación del listado del programa.

Ingresando las líneas del listado 1 podrán tener el juego totalmente completo y empezar a derrochar los viejos $a allí. Buena suerte

LISTADO NUMERO 1

```
9 FR$(1)="{RED}{RVON} "+CHR$(193)+" ":FR$(2)="{CYAN}{RVON} "+CHR$(218)+" "
10 FR$(3)="{YELO}{RVON} "+CHR$(216)+" ":FR$(4)="{ORNS}{RVON} "+CHR$(211)+" "
11 AR$=CHR$(207)+CHR$(183)+CHR$(183)+CHR$(183)+CHR$(208)+"     "
12 ME$=CHR$(180)+"     "+CHR$(170)+"     "
13 AB$=CHR$(204)+CHR$(175)+CHR$(175)+CHR$(175)+CHR$(186)+"     "
22 FORI=1TO4:FE$(I)=FR$(I):NEXTI
1040 F=9:C=0:GOSUB100:FORI=0TO39:PRINTCHR$(164);:NEXTI
1050 F=11:C=3:GOSUB100:PRINTCHR$(207);
1060 FORI=1TO19:PRINTCHR$(183);:NEXTI
1070 PRINTCHR$(208)
1080 F=12:GOSUB100:PRINTCHR$(180);"SU SALDO: $A          ";CHR$(170)
1090 F=13:C=3:GOSUB100:PRINTCHR$(204);
1100 FORI=1TO19:PRINTCHR$(175);:NEXTI
1110 PRINTCHR$(186)
1120 F=15:C=3:GOSUB100:PRINTCHR$(207);
1130 FORI=1TO31:PRINTCHR$(183);:NEXTI
1140 PRINTCHR$(208)
1150 F=16:GOSUB100:PRINTCHR$(180);"                              ";CHR$(170)
1160 F=17:C=3:GOSUB100:PRINTCHR$(204);
1170 FORI=1TO31:PRINTCHR$(175);:NEXTI
1180 PRINTCHR$(186)
1190 F=19:C=3:GOSUB100:PRINTCHR$(207);
1200 FORI=1TO20:PRINTCHR$(183);:NEXTI
1210 PRINTCHR$(208)
1220 F=20:GOSUB100:PRINTCHR$(180);"SU GANANCIA: $A        ";CHR$(170)
1230 F=21:GOSUB100:PRINTCHR$(204);
1240 FORI=1TO20:PRINTCHR$(175);:NEXTI
1250 PRINTCHR$(186):S=1000:G=0
```




INTERCAMBIO DE PROGRAMAS

Primero que todo quisiera disculparme por crear que la tardanza de los ejemplares era de su parte.
Las revistas son magníficas, no cabe duda de que su esfuerzo por lanzarlas al mercado no ha sido en vano, debido a que son un verdadero éxito.
Junto con esta carta les envío el programa para el concurso. Este programa no es nada extraordinario porque soy un "principiante", pero mi deseo no es ganar sino participar para integrarme al mundo de la computación. Lo hice en una computadora TI-99/4A del instituto al cual concurro, ya que no poseo una.
También les mando el cupón del sorteo mensual (espero que acepten una fotocopia de él, porque no deseo recortar la página).
Desearía que publicaran que quiero intercambiar programas varios, mi nombre y dirección:

GONZALO VILLAFAÑE
Alberdi 457
San Francisco
(2400) CORDOBA

K-64

Gracias Gonzalo por tu aliento y tu colaboración. Con respecto a los cupones, no hay problema en enviar fotocopias.

ASSEMBLER

Quiero decirles que su revista es muy interesante, ya que hace rato que estoy buscando alguna publicación de este tipo y ésta, por lo que vi hasta ahora, es la que más me gustó. El artículo "Conociendo las Computadoras" está muy bueno. Quiero felicitarlos también por el programa "Práctica de paracaidismo", que funciona muy bien.

Quisiera que publiquen o me envíen si les es posible información sobre cómo se usa el lenguaje Assembler y cuáles son sus ventajas y desventajas, y cómo se implementa y usa en un computador TI-99/4A.

Dario Tamagnini
San Francisco - Córdoba

K-64

El uso de Assembler en cualquier computadora, permite la explotación al máximo de su capacidad operativa.

Sin embargo se requieran profundos conocimientos sobre el tema y sobre el funcionamiento interno de la computadora, como así también la ayuda de programas especiales.

Lamentablemente para la TI99/4A no se dispone de facilidades para ello, debido a que se trata de una máquina que se dejó de fabricar en Estados Unidos y por lo tanto no se recibe más ayuda ni software adecuado para profundizar en este lenguaje de programación.

INTERFERENCIA

Los felicito por vuestra revista y a la vez consulto si conocen que se comercialice un teclado tipo profesional para la TS 1000, o que sea posible armarlo con teclas independientes. También creo que sería bien aceptado un artículo sobre la forma de solucionar la interferencia de R.F. que producen en la pantalla del televisor (no inestabilidad vertical) las TS 1000 y TS 1500?

K-64

No sabemos de nadie que comercialice teclados de ese tipo.
Si se puede sin embargo, adaptar teclados de rezago, de una computadora antigua, o armarlo a partir de teclas individuales de máquina de calcular. Pero es de esperar que al hacer las cuentas del material y tiempo insumido en fabricarlo, no se justifique realmente.
El problema de la interferencia que sale en la pantalla es bastante complejo de resolver, pero aquí van algunos consejos.

- Cambiar el cable coaxil por otro más largo (puede usarse un buen cable blindado tipo micrófono), e ir acortando su longitud hasta lograr mejor imagen.
- Enrollar el cable en una barrita de ferrite (algunas vueltas).
- Idem en un anillo de ferrite.
- Usar otro adaptador de coaxil/300Ω, o directamente probar sin él.
- Tratar de llevar la salida del modulador a otro canal, desatornillando el núcleo de la bobina del mismo.
- Hay televisores que se ven fatal (algunos NOBLEX, por ejemplo).
- Soldar dentro del modulador, en el conector de salida, un capacitor de bajo valor (probar con 47PF).
- Obviar el modulador y parte del TV, conectándolos directamente por video.

¡Esperamos que alguna le sirva!

TS2068: LA INTERFASE 1

Me gustaría saber si la interface "ZX 1" para la conexión de "Microdrives" de la computadora Spectrum es compatible con la computadora TS 2068. Además tengo la duda de si un modem se conecta directamente en las conexiones de la computadora Spectrum o hace falta alguna interface. Una vez conectado la Spectrum puede comunicarse con una Texas TI99/4A, con una Apple o con una IBM, para acceder a información intercambio de programas, etc.

Néstor Hugo López Cabanillas
LOMAS DE ZAMORA
Buenos Aires

K-64

La interface 1 no es compatible con la TS 2068. Además, por lo que sabemos no estará disponible a la venta por algunos meses. Por otro lado, disponiendo de una ZX Spectrum con la Interface 1, se la puede conectar cualquier modem que tenga entrada RS-232 y su Software adecuado para controlarlo. También existen Modems que no necesitan interfaces y se conectan directamente en el conector postarior. En cualquiera de los casos, es posible conectarse con cualquier otra computadora que tenga conectado un Modem que trabaje en la misma norma, (Bell 103 ó CCITT), como así también comunicarse con Bases de datos internacionales.

PROGRAMA PARA TELEGRAFIA

De algunas cartas enviadas a esta redacción por lectores que han tenido dificultades con el programa para telegrafía para computadores TS1000, creemos necesario tener en cuenta los siguientes factores:

a) Para los que no utilizan un programa Ensamblador, en el ingreso del programa, es decir que "Pokean" los códigos del lenguaje de máquina directamente, tener en cuenta que, antes de ingresar los códigos se debe generar una instrucción 1 REM con tantos espacios como códigos se vayan a tipear, pues caso contrario el intérprete BASIC se confundiría y se perdería el control de la máquina, debiéndosele apagar para retomarlo.

b) El programa ha sido chequeado en la práctica durante bastante tiempo y aunque suene redundante FUNCIONA BIEN, siempre que se tengan en cuenta su rango de acción y limitaciones, es decir:

1) La señal recibida debe ser MUY FUERTE y estar por encima del ruido de fondo en por lo menos 30 dB.
2) La transmisión debe ser razonablemente bien hecha en cuanto a calidad de manipulación.
3) El nivel de audio proveniente del receptor debe

SORTEO - ENCUESTA K64

LLENE ESTE CUPON Y PARTICIPE DEL SORTEO MENSUAL

PREMIOS:

20 CASSETTES Y 10 BECAS PARA CURSOS BASIC

NOMBRE .... Edad ....

Domicilio .... TE: ....

C.P. .... Localidad ....

Enviarlo a:
K64 Computación Para Todos
Cerrito 1320 1° (1010) Buenos Aires REP. ARGENTINA

ENCUESTA

COMPUTADORA: ☐ CZ 1000 ☒ CZ 1500 ☐ CZ 2000 ☐ TIMEX 2068 ☐ TK 83
☐ TK 85 ☐ TK 90 ☐ C-16 ☐ C-64 ☐ TI 99/4A
☐ Otras .... ☐ NO TENGO AUN

ME GUSTARIA VER:

MAS	IGUAL	MENOS	
☐	☐	☒	PROGRAMAS DE APLICACION ESPECIFICA
☒	☐	☐	PROGRAMAS EN BASIC
☐	☐	☒	PROGRAMAS EN LOGO
☐	☒	☐	PROGRAMAS EN LENGUAJE DE MAQUINA
☒	☐	☒	PROGRAMAS EN OTROS LENGUAJES
☐	☐	☐	ANALISIS DETALLADOS DE LOS PROGRAMAS
☐	☒	☐	NOTAS PARA BEGGINERS
MAS	IGUAL	MENOS	
☒	☐	☐	JUEGOS
			CALIFICACION DESCRIPTIVA DE:
☒	☐	☐	PROGRAMAS DE JUEGO
☐	☐	☒	PROGRAMAS DE APLICACIONES COMERCIALES
☐	☐	☒	PROGRAMAS EDUCATIVOS
☒	☐	☐	HARDWARE

QUE ES LO QUE MAS TE GUSTA DE K64?

QUE ES LO QUE MENOS TE GUSTA?

ser muy alto requiriendose entre 15 y 20 VPP para el óptimo funcionamiento

4) El tono de la señal debe ser agudo, mucho más del que se utiliza para tomar telegrafía a "oído", algunos receptores tendrán inconvenientes para proporcionar este tono cumpliendo 1) y 3)

c) El nivel que se obtiene como salida en transmisión del conector MIC es exiguo (algunas decenas de milivoltios) y no podrá en la mayoría de los casos excitar adecuadamente el micrófono de un tranceptor de BLU, debiéndose para ello utilizar un pequeño amplificador de audio.

d) La utilización del programa con un tranceptor de BLU no genera señales de tipo R2 (ilegales) sino RI que son perfectamente legales, no obstante es conveniente cuando el uso del programa exceda la corta experimentación el proceder al control de la manipulación mediante un rectificador de audio y la clavija KEY del emisor.

e) El programa ayuda a la introducción en la materia, pero no implica que para un trabajo a largo plazo no se deba aprender Morse por los métodos tradicionales, en tal eventualidad el mismo es un excelente auxiliar.

f) No en todos los receptores de TV se obtiene una nota limpia por el parlante de audio y a modo de monitor, en algunos esto ocurre sin inconvenientes, en otros es necesario regular el control de sintonía y para que la nota emitida sea limpia la imagen de video es mala, y en otros casos (ciertos televisores color) independientemente de lo que se haga no hay forma de lograr sonido alguno.

COMPATIBILIDAD

Tengo una computadora microdigital TK 2000 y quiero preguntarles qué software de otras marcas son compatibles con la mía; ya sean juegos o de otras aplicaciones.

Diego Verruno
Témperley · Pcia. Bs. As

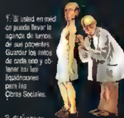
K-64

Diego, la TK 2000 no tiene mucha compatibilidad con otras marcas. Sólo podrían funcionar algunos cassettes de computador Apple.



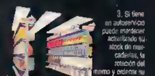
Plá
LS 10
TU CONEXION CON LA MUSICA.
RADIO
Del Plata
LS 10
1030AM/95.1FM ESTEREO.

Sepa exactamente cómo una computadora CZ Spectrum puede prolongar su cerebro.



1. Si usted es médico puede llevar la agenda de turnos de sus pacientes. Guardar los datos de cada uno y obtener así las liquidaciones para las Obras Sociales.

2. Si tiene una Agencia de Prode, quiniela y loterías, puede registrar la venta total de la semana, cuáles son los números más pedidos, los atrasados y los ganadores del Prode. Con una CZ SPECTRUM tendrá la grande en su local.



3. Si tiene un autoservicio puede mantener actualizado su stock de mercaderías, la rotación del mismo y ordenar su facturación.

4. Si tiene una agencia de autos, podrá llevar un registro completo de cada unidad, color, modelo, kilometraje y hasta modificar el precio al instante.

5. Si es constructor podrá tener rápidamente un panorama completo de cada obra. La compra de materiales por rubro, cálculos estadísticos, cantidad de personal, índice de rentabilidad. Con la CZ SPECTRUM, comienzan hoy los cimientos de sus obras del año 2000.

6. Si administra consorcios, ingrese los gastos de cada edificio y obtenga en instantes el cálculo de las expensas.

7. Si vende pasajes podrá tener un control absoluto de todos los vuelos. Teniendo en cuenta los distintos tramos. Compañías. Tarifas en dólares y la transformación automática quincenal y mensual.

8. Si administra clubes, lleve el registro de socios.

9. Si tiene un restaurante, registra fácilmente los cambios de la lista de precios, ingrese los consumos por mesa y obtenga las adiciones a pedido.

10. Si tiene una inmobiliaria, lleve el registro de las propiedades en venta. Hasta que suma puede estudiar una oferta. Qué zona le falta cubrir para ofrecer. Dónde están las propiedades. Qué características tienen. Ubicación, acceso, comodidades, precio.

11. Si su actividad es el comercio exterior, con la CZ SPECTRUM tenga en segundos datos fundamentales para su negocio; cantidad de toneladas cosechadas; precio del grano en los distintos mercados, cantidad de cabezas ingresadas a Liniers en los últimos años; su disponibilidad de mercaderías exportables al instante. CZ SPECTRUM inclina la balanza a su favor.

12. Con la CZ SPECTRUM, un aventajado estudiante como usted, resolverá rápidamente cálculos, ecuaciones o fórmulas y tiene a su alcance cientos de datos de cualquier materia. Y luego, para despejar la mente, corra una carrera de Fórmula 1, rechace un ataque extraplanetario o simplemente dé un paseo en su avión. CZ SPECTRUM la simplificación del estudio.

13. Y cualquiera sea su actividad, con la CZ SPECTRUM, la computación empieza a formar parte de su vida de todos los días para hacerla más fácil.

Porque eso es la tecnología CZ: computación al alcance de todos.



Spectrum

La prolongación de su cerebro

Adquierala en los Distribuidores de la



Distribuye en todo el país: CZERWENY ELECTRONICA S.A. Av. de Mayo 963 - 3° P (1084) Capital Federal